



**FCTUC** DEPARTAMENTO  
**DE ENGENHARIA INFORMÁTICA**  
FACULDADE DE CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA

Mestrado em Engenharia Informática

**Dissertação/Estágio**

Relatório Final

# Remodelação e Desenvolvimento de um Sistema de Informação para ANET

Jennifer Rocha do Carmo Lopes dos Santos

[jennifer@student.dei.uc.pt](mailto:jennifer@student.dei.uc.pt)

2006127498

12 De Julho de 2011

Orientadora do estágio – (Empresa - MagicBrain)

Carina Patrício

Co-orientador do estágio – (DEI)

Luís Silva

[luís@dei.uc.pt](mailto:luís@dei.uc.pt)

## **Agradecimentos**

Devido a contributos de carácter distinto quero aqui expressar os meus sinceros agradecimentos:

Aos meus pais e irmãos que apesar da distância sempre me apoiaram com amor, carinho e amizade.

As minhas avós, em especial a Avó Lica que amo muito e que já não esta entre nós.

Aos meus orientadores de estágio, Carina Patrício, Jorge Oliveira e Professor Dr. Luís Silva, pela orientação no projecto, disponibilidade e paciência durante o estágio.

Aos responsáveis e funcionários da MagicBrain pela confiança, amizade e partilha de conhecimentos.

A todos os meus colegas do curso e amigos, em especial aos meus colegas de noitadas de estudo e ao meu sempre amigo Paulo Ricardo, pela convivência e amizade que tivemos durante todos estes anos.

Muito obrigada a todos!

## Resumo

O presente documento descreve o estágio realizado pela estagiária, que foi integrado num projecto de estudo e desenvolvimento de uma plataforma já existente e em produção.

A implementação inicial deste sistema de informação partiu da experiência do orientador inicial Jorge Oliveira, ao desenvolver uma plataforma na ANET- Associação Nacional dos Engenheiros Técnicos.

O sistema de informação actual da ANET é constituído por 9 módulos: Membros, Inscrição/Estágio, Secretaria, Declarações, Documentação, Meu INETSYS, Contabilidade, Administração e Manuais/FAQ. Apesar da plataforma se encontrar em funcionamento, apresenta algumas limitações, em particular diferenças de comportamento entre browsers e apresentação de redundância nos dados presentes na Base de Dados do sistema. Para contornar tais problemas a estagiária realizou um estudo sobre a plataforma existente, reestruturou e reaplicou a normalização para algumas tabelas da Base de Dados, remodelou e desenvolveu uma nova plataforma com recurso a novas tecnologias e efectuou estudos de performance, *benchmarking*, escalabilidade e segurança na aplicação desenvolvida.

Dada a extensão dos módulos do sistema objecto de estudo e a limitação de tempo imposta pela duração do estágio, tendo como objectivo último o desenvolvimento de uma aplicação estruturada que respondesse positivamente a testes de performance e segurança, a estagiária ficou responsável pela implementação de alguns daqueles.

O trabalho desenvolvido numa primeira fase, coincidente com o primeiro semestre, consistiu no levantamento de requisitos, na análise do desempenho da plataforma em produção na instituição, na análise das soluções tecnológicas e metodologias de desenvolvimento de software a adoptar para o desenvolvimento da nova aplicação.

Na segunda fase, portanto segundo semestre, os trabalhos definidos e realizados pela estagiária centraram-se no estudo de performance (análise de *front-end/back-end*, recorrendo a ferramentas por forma a verificar o comportamento da aplicação), na implementação e reestruturação dos módulos para que se obtenha um bom desempenho ao nível da performance e da segurança, e na avaliação da aplicação mediante a realização de testes ao nível da escalabilidade, segurança, *benchmarking* e *tuning*.

# Índice

<b>1</b>	<b>Introdução .....</b>	<b>1</b>
1.1	Âmbito do Relatório .....	1
1.2	Enquadramento / Objectivos .....	1
1.3	Metodologia de desenvolvimento .....	3
1.4	Estrutura do relatório .....	3
1.5	Documentos desenvolvidos.....	4
<b>2</b>	<b>Descrição da Plataforma .....</b>	<b>7</b>
2.1	Descrição da ANET-PP: .....	7
2.2	Alterações feitas na plataforma ANET-PP .....	8
<b>3</b>	<b>Requisitos Funcionais e Não funcionais .....</b>	<b>9</b>
3.1	Requisitos funcionais:.....	9
3.1.1	<i>Módulo Membros:</i> .....	9
3.1.2	<i>Módulo Insc./Estágios:</i> .....	10
3.1.3	<i>Módulo Documentos:</i> .....	13
3.2	Requisitos não funcionais.....	14
3.2.1	<i>Portabilidade:</i> .....	14
3.2.2	<i>Segurança:</i> .....	14
3.2.3	<i>Performance:</i> .....	16
3.2.4	<i>Escalabilidade:</i> .....	18
<b>4</b>	<b>Estado da Arte .....</b>	<b>21</b>
4.1	Tecnologias Servidores .....	21
4.2	Tecnologias do lado Cliente.....	22
4.3	Outras técnicas utilizadas .....	23
4.4	Ferramentas utilizadas .....	24
4.5	Tecnologias escolhidas para o desenvolvimento do SI e a razões .....	24
4.5.1	<i>Problemas a resolver</i> .....	26
4.5.2	<i>Melhorias aos problemas:</i> .....	26
<b>5</b>	<b>Planeamento e Gestão do Projecto .....</b>	<b>28</b>
5.1	Metodologia usada para o desenvolvimento do Projecto .....	28
5.1.1	<i>Fases do SCRUM</i> .....	28
5.1.2	<i>Artefactos do SCRUM</i> .....	29
5.2	Equipa do projecto .....	33
5.2.1	<i>ProductOwner e ScrumMaster</i> .....	33
5.2.2	<i>ScrumTeam</i> .....	33

5.3	Plano, execução e controle do Sprint.....	33
<b>6</b>	<b>Arquitectura da Aplicação .....</b>	<b>35</b>
6.1	Camada de Apresentação ou Interface .....	35
6.2	Camada de Negócio.....	35
6.3	Camada de Dados .....	36
6.4	Estrutura geral da aplicação .....	36
6.4.1	<i>Aplicação ANET.....</i>	<i>36</i>
6.4.2	<i>Módulos da Aplicação ANET.....</i>	<i>37</i>
6.4.3	<i>Estrutura do código desenvolvida para Aplicação: .....</i>	<i>37</i>
<b>7</b>	<b>Testes .....</b>	<b>38</b>
7.1	Testes funcionais .....	38
7.2	Teste de Escalabilidade .....	39
7.3	Testes de Performance .....	40
7.4	Teste de Segurança.....	41
7.5	Após execução de testes .....	44
7.5.1	<i>Testes.....</i>	<i>45</i>
<b>8</b>	<b>Plano de Trabalho .....</b>	<b>46</b>
<b>9</b>	<b>Conclusões.....</b>	<b>50</b>
<b>10</b>	<b>Referências .....</b>	<b>52</b>

## Definições e Acrónimos

Acrónimo	Descrição
<b>ANET</b>	Associação Nacional dos Engenheiros Técnicos
<b>BD</b>	Base de Dados
<b>SGBD</b>	Sistema de Gestão de Bases de Dados
<b>PDO</b>	PHP Data Objects
<b>PHP</b>	PHP: HypertextPreprocessor
<b>HTML</b>	HypertextMarkupLanguage
<b>PhpMyAdmin</b>	Interface web para a base de dados.
<b>SQL</b>	StructuredQueryLanguage
<b>CSS</b>	CascadingStyleSheets
<b>SRC</b>	Secção Regional Centro
<b>CDN</b>	Conselho Directivo Nacional
<b>CD</b>	Comissão Directiva
<b>CE</b>	Comissão de Estágios
<b>ANET – PP</b>	Plataforma em produção na ANET
<b>ANET-PDE</b>	Plataforma desenvolvida pela estagiária



## **Lista de Figuras**

Figura 1 - Resultado de testes .....	20
Figura 2- Metodologia SCRUM.....	28
Figura 3 - Sprint Semanais.....	30
Figura 4 - TrackChart.....	31
Figura 5 - BurndownChart .....	32
Figura 6 - Arquitectura da aplicação.....	35
Figura 7 - Decomposição do sistema ANET em módulos. ....	36
Figura 8 - Decomposição da aplicação ANET em módulos.....	37
Figura 9 - Estrutura da Aplicação.....	37
Figura 10 - Diagrama de Gantt inicial 1º Semestre – 10 Setembro .....	46
Figura 11 - Diagrama de Gantt inicial 2º Semestre – 7 Fevereiro .....	47
Figura 12–Diagrama de Gantt Final 2º semestre - Junho .....	48

## **Lista de Ilustrações**

Ilustração 1 - Regras do YSlow.....	17
Ilustração 2 - Avaliação ANET-PDE .....	18
Ilustração 3 - Classificação ANET-PDE antes das melhorias .....	41
Ilustração 4 - Classificação da ANET-PDE depois das melhorias.....	41

## **Lista de Tabelas**

Tabela 1 - Parâmetros de teste de Escalabilidade.....	20
Tabela 2 - Estrutura geral da aplicação .....	36
Tabela 3 - Resultados de testes de escalabilidade.....	40
Tabela 4 - Resultado de testes de segurança .....	44
Tabela 5 - Erros encontrados/melhorados.....	44



# **1 Introdução**

Neste capítulo é referido o âmbito do estágio, sendo descrita a empresa onde o mesmo foi realizado. Serão focados os objectivos quer do projecto quer do estágio em particular, referindo onde o mesmo se enquadra e a metodologia de desenvolvimento. E por último segue uma pequena descrição da estrutura do documento.

## **1.1 Âmbito do Relatório**

Este estágio encontra-se inserido na disciplina de Estágio/Dissertação do Mestrado em Engenharia Informática do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologias da Universidade de Coimbra e foi realizado na empresa MagicBrain – OFPG Lda, em Coimbra, sob a orientação de Carina Patrício (nesta empresa), e do Professor Dr. Luís Silva (no Departamento de Engenharia Informática).

A MagicBrain – OFPG Lda, é uma empresa com sede em Coimbra, e foca os seus produtos na área de desenvolvimento e segurança de aplicações Web.

O estágio teve início a 6 de Setembro de 2010, em regime part-time, durante o primeiro semestre, com presença na empresa nos dias úteis entre as 10h as 14h. E em regime full-time no 2 semestre, do dia 7 de Fevereiro a 30 de Junho, entre as 10h as 19h.

## **1.2 Enquadramento / Objectivos**

Hoje em dia existe um crescente volume de utilização da internet e um forte aumento a nível de expansão de aplicações Web, sendo um dos principais factores para o sucesso ou insucesso de uma aplicação Web a forma como ela se comporta (o seu desempenho).

A ANET é uma associação pública representativa dos detentores do bacharelato em Engenharia que disponibiliza vários serviços administrativos no actual Sistema de Informação.

Este sistema apresenta alguns problemas, os quais dificultam o seu bom funcionamento e o acesso por parte dos utilizadores à informação que consta nos serviços desse Sistema de Informação. Por esse motivo para a referida associação existiu a necessidade de remodelar e desenvolver uma nova plataforma que suprissem aquelas dificuldades.

O estágio surge então integrado na ideia da estagiária desenvolver um projecto, cujo objectivo consiste no estudo e desenvolvimento de uma nova aplicação Web para a ANET de forma agilizar o seu funcionamento para os seus utilizadores. Assim, o desenvolvimento do projecto visa alcançar, principalmente, os seguintes objectivos:

- Reestruturação da Base de Dados.
- Reestruturação da aplicação, garantindo uma compatibilidade entre browsers, para que os utilizadores não tenham que estar limitados a um em particular.
- Realizar estudos da performance, escalabilidade e segurança da aplicação, por forma garantir uma aplicação eficiente, robusta e flexível.

Para que o projecto possa responder afirmativamente aqueles objectivos foi requerido à estagiária a realização das seguintes tarefas:

- Elaboração do caso de estudo para a aplicação Web ao nível de performance e de funcionalidades da mesma.
- Escolha de ferramentas que melhor se adaptem para o estudo e análise do sistema de informação.
- Escrita de um documento com uma análise global no qual sejam identificadas as principais falhas da aplicação Web, apresentando para as mesmas possíveis soluções.
- Escrita de um documento com os requisitos funcionais a implementar.
- Análise das soluções tecnológicas a adaptar.
- Escolha da metodologia de desenvolvimento de software que melhor se adapta ao projecto.
- Implementação dos módulos segundo a metodologia de desenvolvimento escolhida, procedendo-se às seguintes tarefas:
  - Levantamento de requisitos.
  - Análise de alterações/melhoramentos da plataforma existente segundo o estudo efectuado.

- Desenvolvimento, testes e validação dos módulos.
- Escrita do documento dos testes funcionais da aplicação Web.
- Avaliação dos requisitos não funcionais a nível de segurança, *benchmarking* e *tuning*, em cenários de múltiplos.
- Escrita de um documento que descreva os testes de performance, segurança e escalabilidade realizados.

### 1.3 Metodologia de desenvolvimento

A escolha de uma metodologia de desenvolvimento recaiu sobre o **SCRUM**, por se apresentar como sendo ágil, incremental e iterativo. O projecto de estágio foi dividido em iterações de 15 dias, denominadas de *Sprints* que representam o tempo definido dentro do qual um conjunto de actividade deva ser executado. As funcionalidades implementadas durante o projecto representam uma lista conhecida com *ProductBacklog*.

O início de cada *Sprint*, coincidia com um *Sprint Plannig Meeting* (reunião de especificação do plano de tarefas) na qual o *ProductOwner* (a orientadora) especificava os itens que fariam parte do *ProductBacklog* que a estagiária teria de desenvolver para aquele período.

O final de cada *Sprint* culminava com um *Sprint Review Meeting*, reunião na qual se verificava o cumprimento dos objectivos estabelecidos para aquele período, a estagiária apresentava as funcionalidades implementadas e os resultados de verificação das mesmas e expunha as dificuldades encontradas. Na mesma reunião fazia-se um *Sprint Retrospective* para identificar o que funcionou bem, o que poderia ser melhorado e que acções realizar.

### 1.4 Estrutura do relatório

O documento encontra-se organizado em vários capítulos, descritos de seguida:

**(Resumo)** – é apresentado de maneira concisa o projecto, âmbito e objectivos.

Capítulo 1 (**Introdução**) – apresenta o âmbito do estágio, enquadramento, objectivos/motivação quer do projecto quer do estágio em particular, a metodologia de desenvolvimento escolhida. E por fim apresenta a estrutura do relatório e os documentos desenvolvidos.

Capítulo 2 (**Descrição da ANET-PP**) – breve descrição da plataforma em produção na ANET e especificação das alterações realizadas aquando do desenvolvimento da ANET-PDE.

Capítulo 3 (**Requisitos funcionais e não funcionais**) – apresentação dos requisitos funcionais e não funcionais.

Capítulos 4 (**Estado de arte**) – apresentação das tecnologias e ferramentas utilizadas para o desenvolvimento da plataforma, sua descrição, apresentação das vantagens da sua utilização e o porquê da escolha daquelas tecnologias e ferramentas.

Capítulo 5 (**Planeamento e Gestão do Projecto**) – descrição do modo como o projecto foi coordenado, a equipa que o constitui, a metodologia utilizada, a divisão das tarefas e a planificação detalhada das mesmas pelo período de estágio.

Capítulos 6 (**Testes**) – descrição dos testes de performance, segurança, escalabilidade, *banchmarking* e *tuning* realizados ao longo da fase de implementação do projecto.

Capítulo 7 (**Plano de Trabalho**) – apresentação do plano do projecto descrevendo o trabalho realizado durante as diferentes fases.

Capítulo 8 (**Conclusão**) – apresentação das conclusões sobre o projecto e das considerações finais para o mesmo.

## 1.5 Documentos desenvolvidos

Foram produzidos diversos documentos, principalmente na fase inicial do projecto, servindo de suporte para uma melhor compreensão do mesmo e uma correcta coordenação de tarefas.

Os vários documentos produzidos encontram-se no CD-ROM entregue juntamente com o relatório.

Em seguida, segue-se uma pequena descrição sobre cada documento elaborado:

- **Anexo A (Caso de Uso)**, descreve os requisitos funcionais necessários para o projecto, apresentando os módulos e respectivos submenus em que o desenvolvimento do projecto foi dividido. Esses casos de uso foram também utilizados como caso de testes.
- **Anexo B (Arquitectura)**, descreve a arquitectura dos vários módulos a implementar, tendo como referência o documento de Casos de Uso. Apresenta também a estrutura do sistema de informação e também uma descrição das implementações realizadas e como aquelas se encontram estruturadas.
- **Anexo C (Modelo ER Módulos)**, descreve o modelo de dados da aplicação bem como as alterações que foram feitas e procedimentos para a realização daquelas alterações.
- **Anexo D (Análise Performance da ANET-PP)**, este documento descreve a análise de performance que foi feita na Plataforma em produção na ANET.
- **Anexo E (Análise Performance da ANET-PDE)**, este documento descreve a análise de performance realizada sobre a aplicação que a estagiária desenvolveu, descrevendo as alterações que foram implementadas para melhorar a performance *Front- End*.
- **Anexo F (Sprints Semanais)**, este documento foi escrito seguindo a metodologia de desenvolvimento adoptada, **SCRUM**, apresentando e descrevendo as tarefas atribuídas semanalmente e a evolução das mesmas. Este documento possui um ficheiro auxiliar **Sprint\_Projecto\_Estagio.sprz** que resulta da ferramenta “*sprintometer*” utilizada para criar os *Sprints* e gráficos seguindo a metodologia **SCRUM**.
- **Anexo G (Testes Funcionais)**, neste documento encontram-se especificados os testes funcionais que foram feitos na aplicação, para verificar se esta funcionava correctamente e se está de acordo com os objectivos estipulados nos *sprints*.
- **Anexo H (Plano Testes e Testes Segurança)**, apresenta o Plano de Testes de Segurança efectuado na aplicação bem como os respectivos resultados, erros obtidos na execução e soluções para a resolução daqueles problemas.
- **Anexo I (Plano Testes e Teste Portabilidade)**, apresenta o Plano de Testes de Portabilidade efectuado na aplicação bem como os resultados obtidos e correcções realizadas nos pós testes.

- **Anexo J (Testes Escalabilidade)**, apresenta os testes de escalabilidade efectuados na página que se pensa que terá maior fluxo de utilização entre os módulos que foram implementados pela estagiária. O documento apresenta também os resultados obtidos e as conclusões tiradas relativamente àquela página.

Este tipo de teste vai ajudar a identificar a capacidade máxima de funcionamento da aplicação, ou seja qual é o número máximo de utilizadores que a aplicação consegue suportar em simultâneo.

## 2 Descrição da Plataforma

Nesta secção será feita uma breve descrição da plataforma ANET-PP de forma a entender melhor a aplicação, apresentando o tipo de serviço que disponibiliza aos seus destinatários. Posteriormente serão apresentados as alterações que foram realizadas com a implementação da nova aplicação.

### 2.1 Descrição da ANET-PP:

A plataforma, cujo acesso acontece mediante *browser* e apenas para os funcionários daquela associação, está direccionada para gestão de conteúdos da ANET disponibilizando os seguintes serviços: Administrativos, Contabilísticos, Financeiros/Fiscais e de Secretariado representados na plataforma pelos módulos abaixo descritos:

- Módulo **Membro**, gestão e pesquisa de informação sobre os membros.
- Módulo **Inscrição/Estágio**, gestão dos processos de estágio, desde a inscrição, período de estágio até às audições finais do processo e gestão dos cursos de Ética.
- Módulo **Secretária**, responsável pela gestão de etiquetas, de comunicação feitas aos membros da associação, pedidos de prorrogações de estágios, consulta de recibos e a criação de SIBS para um dado membro.
- Módulo **Declaração**, gestão das declarações existentes, carregamentos de contas e visualização dos movimentos.
- Módulo **Documentação**, gestão de correspondências (correio ou email) e de documentação.
- Módulo **InetSys**, gestão da informação pessoal dos utilizadores e da informação de sistema.
- Módulo **Contabilidade**, responsável pela gestão contabilística da associação.
- Módulo **Administração**, responsável pela gestão de utilizadores, site institucional da associação, actualização de legislação e actualização da BD.
- Módulo **Manuais/FAQ**, módulo informativo para o auxílio dos utilizadores.

## 2.2 Alterações feitas na plataforma ANET-PP

Como já referido anteriormente a ANET-PP apresenta 9 módulos e dado ao elevado número de módulos foram implementados apenas quatro daqueles módulos, de modo que a estagiária pudesse realizar uma melhor análise e validação dos mesmos.

A ANET-PP apresentava alguns problemas de incompatibilidade entre os browsers mais utilizados e redundância de dados na Base de Dados. Para colmatar aquelas falhas, a estagiária recorreu a um conjunto de tecnologias e ferramentas que achou capazes de produzir um produto final eficiente.

### **A ANET-PDE apresenta as seguintes diferenças em relação à ANET-PP:**

- Os menus da ANET-PP foram implementados em PHP e *JavaScript*, o que veio a resultar numa das maiores causas de incompatibilidade verificada entre *browsers*, uma vez que cada *browser* possui o seu *standard* para *JavaScript*. Para a ANET-PDE, o menu foi implementado recorrendo exclusivamente ao PHP cuja interpretação não varia com o *browser*.
- Algumas das funcionalidades existentes foram reestruturadas para acompanhar a própria evolução da associação também novas funcionalidades foram acrescentadas.
- Apresenta um *layout* reestruturado.
- A ligação à Base de Dados acontece recorrendo à extensão do PHP5, PDO, o que traz vantagens ao nível de segurança promovendo a protecção ao *SQL Injection* mediante ligações persistentes, transacções e preparação (*prepare*) de consultas (*query*).
- A base de dados foi reestruturada recorrendo ao processo de normalização por forma combater problemas de redundância.



## 3 Requisitos Funcionais e Não funcionais

Nesta secção serão apresentados os requisitos funcionais relativamente aos módulos implementados pela estagiária.

### 3.1 Requisitos funcionais:

Os requisitos funcionais descrevem as interações do sistema com o seu ambiente, ou seja são aqueles que descrevem o conjunto de funcionalidades que se pretendem ver no produto final, os serviços que o sistema deve prover, o comportamentos do sistema a determinadas entradas.

As funcionalidades implementadas na plataforma desenvolvidas pela estagiária serão descritas e apresentadas em seguida, sendo que a descrição com as devidas condições de uso encontram-se no documento “Anexo A”, e a arquitectura e estruturação da plataforma (organização dos módulos) estão no documento “Anexo B”.

#### 3.1.1 Módulo *Membros*:

O módulo membros é constituído pelos seguintes sub-menus:

**1 Listagem Membros**, com as seguintes opções:

- 1.1 Membro com fotos** – apresenta uma listagem dos membros com fotos.  
E uma opção de pesquisa rápida.
- 1.2 Membro sem fotos** – apresenta uma listagem dos membros sem fotos.  
E uma opção de pesquisa rápida.

**2 Pesquisa**, com as seguintes opções:

- 2.1 Pesquisa Directa** – caracteriza-se pela presença de campo único de pesquisa, o utilizador insere um termo e especifica o critério que pretende que aquele termo satisfaça, por exemplo nome de membro, número de contribuinte.
- 2.2 Pesquisa Composta** – a pesquisa acontece mediante uma combinação de opções escolhidas e termos inseridos pelo utilizador.

- 3 **Gestão de Formações** – Devolve uma lista de formações em que cada uma dessas formações tem associadas quatro acções:
  - Visualização do número de inscritos para a formação seleccionada. Com as opções de editar, visualizar e eliminar um determinado membro da listagem.
  - Visualização/Exportação daquela listagem para Excel.
  - Visualização/Exportação daquela listagem para PDF.
  - Inscrição, o utilizador tem a oportunidade de inscrever um membro naquela formação/curso.
- 4 **Lista Membros CDN** – apresenta a listagem de membros pertencentes à SRC, separando-os pelos respectivos grupos: estagiários, estudantes, efectivos, em que para cada um desses grupos estão associadas duas acções:
  - Visualização da lista dos membros daquele grupo.
  - Exportação para PDF da lista dos membros daquele grupo.
- 5 **Competências** – Apresenta um formulário de pesquisa para visualizar as competências de um dado membro, assim como informação pessoal do membro pesquisado.
- 6 **Planear Formações** - Apresenta uma listagem dos cursos de formação, que a SRC está a prever implementar para o ano, disponibilizando aos membros uma pré-inscrição de forma a avaliar a viabilidade de cada um. Para cada uma das formações são apresentadas as opções:
  - Visualizar, lista dos membros que se inscreveram numa determinada formação, com informações relevantes desse membro.
  - Abrir, hiperligação disponibilizada para que após viabilizar a formação (mediante o número de inscrições registadas) possa efectivamente passar o estado daquela formação a “**aberta**” e assim figurar na listagem de formações no sub-menu *Gestão de Formações*.

### 3.1.2 Módulo Insc./Estágios:

Esse módulo apresenta as seguintes funcionalidades (sub-menus):

#### 1 **Gestão de Inscrições:**

##### 1.1 **Menu com links para:**

##### 1.1.1 **Lista de audições Mensal;**

**1.1.2** Lista de relatório de estágio mensal;

**1.1.3** Lista de estágios mensal;

Cada um apresentando um documento PDF gerado com dados da BD.

**1.2 Inscrições Pendentes:**

Apresenta a lista de inscrições em estágios que se encontram pendentes. Sobre esta listagem, pode-se realizar um conjunto de operações: visualizar, editar e eliminar dados dos membros da lista.

**2 Parecer,** apresenta as seguintes opções:

**2.1 Parecer da CE**

Listagem dos inscritos em estágios por aprovar (formais/curriculares) com a opção de ver respectivos dados.

**2.2 Parecer do CD da SRC**

**2.2.1** Listagem dos estágios aprovados (formais/curriculares).

**2.2.2** Listagem das prorrogações aceite.

**2.2.3** Listagem dos Pedidos de Alteração de Patrono com as respectivas entidades de acolhimento.

**3 Comunicação ao Membro,** apresenta listagens com:

- Comunicações ao Membro – Curriculares.
- Comunicações ao Membro - Formais.
- Listagem das últimas comunicações enviadas.

**4 Audições,** com as seguintes opções:

**4.1 Marcação de Audições**

Lista os membros com possibilidade de marcar uma data de audição.

**4.2 Resultado de Audições**

Lista as audições por concluir e lista as audições concluídas. Apresenta a opção gerar PDF/Excel para a lista de audições concluídas num determinado período.

**5 Curso de Ética,** apresenta as seguintes opções:

- Apresenta uma lista das turmas, em que cada turma tem associado duas opções:

- Abrir, se a turma possuir membros inscritos. Após a abertura de turma, é possível ao utilizador listar os membros da turma, alterar dados da turma e imprimir a lista dos inscritos na turma (PDF);
- Inscrever, associar membros à turma;
- Opção “adicionar turmas” e uma pesquisa rápida.

## **6 Cursos de Ética realizados**

Lista os cursos de éticas realizadas, com opção de impressão dos detalhes do curso, nomeadamente membros inscritos.

## **7 Estágios, com as seguintes opções:**

### **7.1 Gestão de estágios - Apresenta um submenu com:**

#### **7.1.1 Acções a executar:**

- Lista de todos os estágios concluídos.
- Lista de estágios em curso (informação dos membros e dos documentos que estes têm em falta).
- Lista de estágios por concluir - com as opções:
  - Inserir informação de documentos entregues.
  - Visualizar a lista de documentos entregues para um dado membro.
  - Anexar documentos aos já existentes.

#### **7.1.2 Módulo de Gestão de Conclusões de Estágios:**

Pesquisa consoante: nome, número definitivo ou número provisório. Para cada pesquisa é apresentada uma lista com as opções:

- Envio de documentação.
- Documentos.
- Estágio - apresenta um formulário para edição dos dados do estágio.

### **7.2 Estágios em atraso**

Lista de membros que já deveriam ter concluído estágio.

## **8 Envio de processos CDN<sup>1</sup>**

Apresenta informação relacionada com o processo e enviar para que seja avaliado pela CDN mediante preenchimento de campo de pesquisa.

## **9 Pedidos de patrono**

Apresenta a listagem com os pedidos de patrono com a opção de visualizar os dados desse patrono e a possibilidade de atribuir patrono.

## **10 Bolsa de patronos**

Lista todos os membros com a bolsa de patrono, com opções de remoção de registos.

### **3.1.3 Módulo Documentos:**

Este módulo apresenta as seguintes opções:

- Entrega de documentos, mediante upload de ficheiros para um dado membro.
- Listagem de documentos, mediante campo de pesquisa, lista os documentos associados ao membro pesquisado, caso seja válido o termo inserido.
- Entrega de documentos em separado, converte os documentos com formato imagem inseridos por upload para formato PDF, por exemplo o BI do membro pode constituir um scan de duas páginas, que convertido nesta funcionalidade anexa aquele conteúdo num único ficheiro PDF.

---

<sup>1</sup> Para este submenu foi apenas implementado o *layout* e a verificação do campo de pesquisa, para que retorne mensagem de erro caso não encontre um resultado válido. O envio não foi implementado pela estagiária

## 3.2 Requisitos não funcionais

Os requisitos não funcionais são requisitos que declaram restrições ou atributos de qualidade para um software e/ou para o processo de desenvolvimento da aplicação. Estes estão normalmente associados a qualidade, desempenho, portabilidade, confiabilidade e segurança da aplicação.

Para o desenvolvimento da aplicação, a estagiária teve o cuidado de se debruçar nos pontos abaixo descritos:

### 3.2.1 Portabilidade:

- Garantir o bom funcionamento da aplicação em todos os ambientes, isto é permitir que a aplicação não tenha limitações nos principais *browsers* (IE, *Chrome*, *Mozilla*, *Opera*, *Safari*), nos diferentes Sistemas Operativos e nas diferentes resoluções de ecrã.

### 3.2.2 Segurança:

Para ter uma aplicação segura a estagiária teve que ter em conta as questões fundamentais da segurança: privacidade, integridade, autenticidade do emissor e *backup* de dados.

- **Privacidade:** Garantir que os dados permaneçam privados e confidenciais (encriptação) e que não possam ser lidos por pessoas não autorizadas, ou seja garantir acesso à plataforma só a utilizadores autorizados. Também garantir um nível de acesso à informação, mediante a definição de diferentes níveis de acessos para cada utilizador, com base nos quais é dado acesso a diferentes áreas da aplicação.
- **Integridade:** Garantir segurança e integridade dos dados inseridos, contra ataques do tipo **XSS** e **SQL Injection**.
  - **XSS-Injection[1]**

Os XSS também conhecidos com CSS (*cross site script*), é um tipo de ataque muito comum normalmente em aplicações *Web*, permitindo ao atacante inserir códigos maliciosos nas páginas executados em determinado momento provocando resultados corrompidos ou afectando as bases de dados que tratam aquela informação.

O ataque permite que conteúdos (*scripts*) de zonas sem privilégios sejam executados com permissão em zonas privilegiadas. Normalmente o tipo de ataque coincide com a inserção de códigos (por exemplo: `<script>alert("XSS")</script>`) em campos de texto que depois de processado resulta num resultado anormal para o funcionamento.

Para proteger contra aquele tipo de ataque, os campos de texto dos formulários foram protegidos com recurso a validações em *javascript/jquery* e a utilização de funções pré-definidas em PHP como o ***htmlspecialchars*** que converte os caracteres especiais no correspondente em HTML (“<” para o “&lt”).

- ***SQL Injection***[2]

*SQL injection* é um tipo de ameaça de segurança em sistemas que interagem com base de dados via SQL. Normalmente este tipo de ataque acontece quando são inseridas instruções SQL em campos de pesquisa ou inserções de dados que irão interferir directamente com a *query* associada ao campo e consequentemente com a Base de Dados, podem ser instruções para eliminar dados ou tabelas ou instruções para conseguir listagens críticas, obter dados de acesso, etc.

Para proteger contra tais ataques foram utilizadas ferramentas disponibilizadas pelo PDO, nomeadamente a aplicação do *bindParam* e do *prepare* nas funções SQL desenvolvidas.

Foram realizados testes nos formulários para verificar se estes se encontravam protegidos contra tais ataques. Os testes encontram-se documentados no “Anexo H”.

- **Autenticidade:** Para se obter segurança ao nível da *Internet* numa aplicação *Web*, várias técnicas são utilizadas, uma delas recorre ao uso do **Protocolo de Segurança SSL** (*SecureSocketLayer*), uma tecnologia de segurança padrão para criar uma ligação encriptada entre um servidor de *Internet* e um navegador de *Internet* (*browser*). Esta ligação assegura que toda a informação trocada entre o servidor e o browser mantém-se privada e integral. Outra técnica utilizada, remete para o uso de teclado virtual na autenticação do utilizador (*username*, *password* e n.º de PIN (inserido via teclado virtual)).
- **Backup de dados:** Promover a segurança dos documentos mediante a realização de *backups* periódicos, preferencialmente num local independente do local original dos documentos, por exemplo num outro servidor externo ou não à instituição, mas sem ligação à Plataforma.

### 3.2.3 Performance:

Pretende-se que a aplicação apresente as seguintes características, com o intuito de responder positivamente a análises de performance:

- Páginas leves e rápidas no carregamento: uma melhoria da performance neste ponto implica uma maior rapidez no tempo de resposta aos utilizadores da aplicação, incrementando a satisfação. Para tal, durante a implementação existiu a preocupação de seguir determinadas regras para garantir uma boa performance. Após a implementação realizou-se uma análise com o auxílio de um avaliador de performance, tendo-se escolhido o “Yslow”[3].
- Rapidez na apresentação dos resultados das pesquisas, procurando garantir que as *queries* para realizar a pesquisa sejam as mais leves possíveis, pelo que houve a necessidade de melhorar algumas sub-consultas[4] utilizadas.

Para melhorar a performance da aplicação desenvolvida foram realizados testes performance seguindo algumas regras de performance segundo **Steve Sourd**[3], com o auxílio da ferramenta “YSlow”. As regras encontram-se classificadas numa escala de A (melhor nota) à F (pior nota). A figura seguinte representa as regras enunciadas.



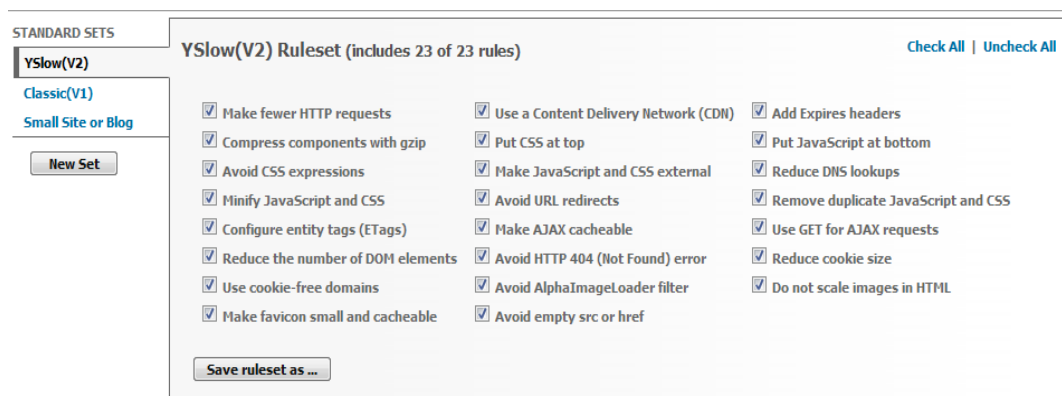


Ilustração 1 - Regras do YSlow

Uma análise inicial à aplicação, revela uma classificação **C** o que significava limitações nas seguintes regras de alta performance:

- *Minimize HttpRequest*, classificação **F**, regra relacionada com o número de pedidos *Http*, tendo-se optado por minimizar o número de imagens;
- *Use a Content Delivery Network (CDN)*, classificação **F**, esta regra não foi aplicada uma vez que a aplicação vai ser usada em Portugal, não havendo a necessidade de criar uma rede de distribuição de conteúdos.
- *Add Expires Headers*, classificação **F**, regra relacionada com os conteúdos estáticos. Para melhorar a sua classificação foi necessário especificar/configurar uma data de expiração, evitando assim que o servidor vá carregando o conteúdo a cada chamada. Com tais melhorias aquela regra passou a ter uma classificação **D**.
- *Minimize JavaScripte CSS*, classificação **F**, para melhorar a sua classificação foi usado a ferramenta **smush.it** (integrada no *YSLOW*) para reduzir o tamanho dos ficheiros, eliminando os caracteres desnecessários, e foram removidos as referências *JavaScriptInline*, tendo sido criado um ficheiro para esse fim (**extras.js**). Com tais melhorias aquela regra passou a ter uma classificação **B**.

- *Configure ETags*, classificação **F**, relacionada com o mecanismo que *Web-Servers* e *Browsers* usam para validar componentes armazenados em cache, para melhorar a sua classificação foi necessário configurar o “*httpd.conf*” para se conseguir guardar os ficheiros em cache, definindo uma data de expiração para os ficheiros que não vão ser alterados num certo período de tempo. Com a configuração e a alteração das regras verificou-se uma melhoria no carregamento dos módulos na aplicação. As alterações ao ficheiro referido encontram-se descritas no documento “Anexo E”.

Com as melhorias realizadas e descritas acima, a classificação da aplicação passou de **C** para **B**, aumentando assim o seu desempenho, avaliando pela figura seguinte:

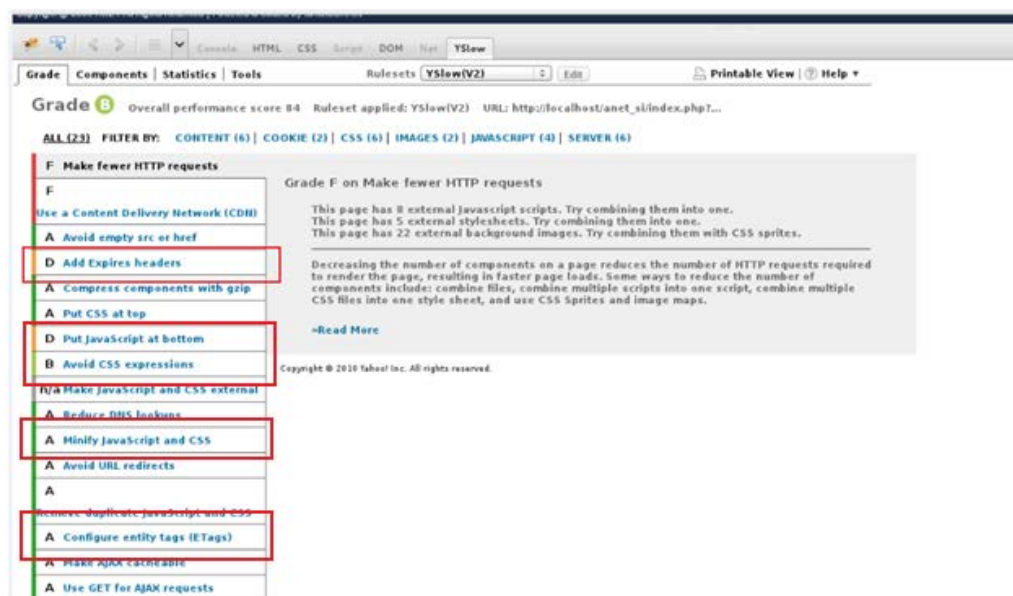


Ilustração 2 - Avaliação ANET-PDE

Este estudo e análise encontra-se com maior detalhe no documentado “Anexo E”.

### 3.2.4 Escalabilidade:

A aplicação deve estar preparada para acessos simultâneos de vários utilizadores, ou seja a aplicação deve-se comportar da mesma forma, ou com uma ligeira diferença a nível de escalabilidade e de performance, mesmo quando o número de acessos aumentar.

Para esse efeito fez-se um estudo da aplicação com o auxílio da ferramenta “*httperf*”[5], ferramenta que mede o desempenho do servidor *Web*. Estes testes foram feitos com o intuito de verificar como a aplicação se comportaria caso o número de utilizadores fosse incrementando. A aplicação será usada somente por funcionários da associação, que pode corresponder a um número máximo de utilizadores que poderá rondar os 100.

### **Página de teste:**

“helios.srcentro-anet.org/anet\_si/index.php?mod=man\_insc\_estag”, sendo essa a pagina que será mais utilizada pelos utilizadores.

### **Parâmetros de teste:**

Parâmetros	Descrição
-- client=0/1	Especifica que a máquina <i>httperf</i> em execução é um cliente, com um total de <b>N clientes</b> , com N a variar normalmente de 0 a N-1. Este parâmetro também é usado para garantir que nem todos os clientes geram cargas de trabalho perfeitamente idênticos.
server	Especifica o endereço ou <b>IP</b> do servidor onde será feito o teste neste caso: helios.srcentro-anet.org
Port	Porto de acesso definido.
Uri	Especifica o <i>URL</i> da página que vai ser sujeita ao teste, no caso: <a href="http://helios.srcentro-anet.org/anet_si/index.php?mod=man_insc_estag">http://helios.srcentro-anet.org/anet_si/index.php?mod=man_insc_estag</a>
Rate	Especifica a taxa fixa na qual as ligações ou sessões são criadas
num-conns	Especifica o número total de ligações para criar
Num-calls	Especifica o número total de chamadas a emitir em cada ligação antes de as mesmas serem fechadas. Se N for maior que 1, o servidor deve suportar ligações

persistentes.

Tabela 1 - Parâmetros de teste de Escalabilidade

Cada teste foi executado em média 5 vezes, tendo sido extraído o pior resultado dos 5 testes efectuados. Abaixo é apresentado o gráfico com os resultados dos testes:

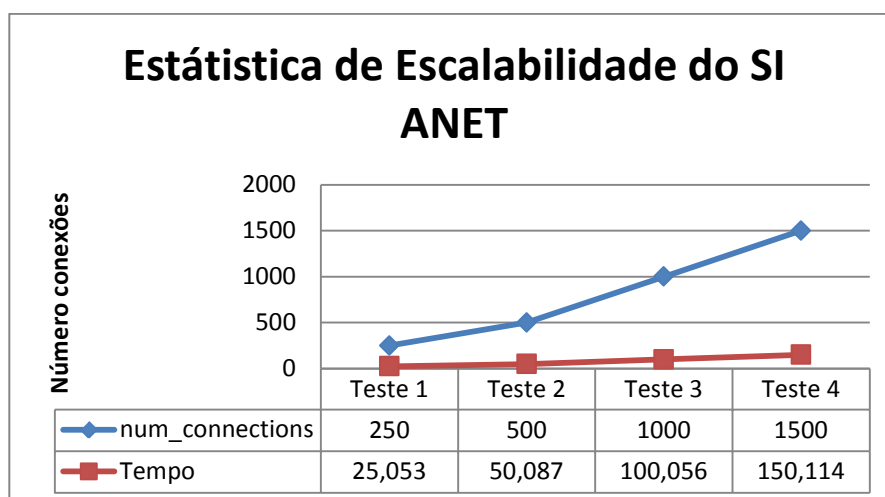


Figura 1 - Resultado de testes

O gráfico a vermelho mostra o tempo médio de resposta para as chamadas de sucesso, como se pode constatar, à medida que se aumenta o número de ligações o tempo de resposta mantém-se, conclui-se que se trata de uma aplicação escalável.

Como referido anteriormente este teste foi realizado para verificar o comportamento da aplicação, e para determinar o rendimento necessário para suportar a carga de pico de produção esperada: uma aplicação do foro administrativa que será utilizada somente pelos funcionários da associação, que rondaram aproximadamente 100 funcionários.

## 4 Estado da Arte

Dado um projecto orientado para desenvolvimento de uma aplicação *Web*, nesta secção serão descritas as tecnologias mais usadas e posteriormente serão especificadas as tecnologias utilizadas, acompanhadas com as justificações da sua escolha.

No desenvolvimento de aplicações *Web* há que ter em consideração as tecnologias usadas do lado do cliente e também do lado do servidor. Essas tecnologias serão descritas a seguir.

### 4.1 Tecnologias Servidores

**PHP (*HypertextPreprocessor*)**[6] é uma linguagem de programação usada para *Web*, tecnologia *open source* de apresentação de páginas *Web* dinâmicas que amadureceu nos últimos anos. Trabalha do lado do servidor e pode ser embutido directamente no código HTML, sendo independente do sistema operativo. Com a versão PHP 5.0, é introduzido um novo conceito, modelação orientada a objectos (PDO) permitindo melhor desempenho e mais vantagens.

**JSP**[7] é um acrónimo para “*JavaServerPage*” que é uma tecnologia que serve para apresentar páginas *web* dinâmicas usando a tecnologia *java* e pode ser considerada uma extensão da tecnologia *Java Servlet*. Por ser baseada na tecnologia *Java*, tem a vantagem da portabilidade de plataforma, que permite a sua execução em diversos sistemas operativos.

**Active Server Pages (ASP)**[8]: *ASP* da *Microsoft* emprega tecnologias de *script* que funcionam em plataformas *Windows*, embora haja esforços para transportar essa tecnologia para outros sistemas operacionais. *ASP Windows* funciona com o servidor *Web Internet Information Server* (IIS – Servidor de Informações de Internet).

**Active Server Pages .NET (ASP .NET)**: Esta tecnologia é parte integrante da *.NET* da *Microsoft*. A *.NET Framework* emprega um tempo de execução chamado *CommonLanguageRuntime* que é muito semelhante a *Java Virtual Machine* e oferece uma vasta biblioteca de classes, disponível a todas as linguagens *.NET* e de *ASP.NET*. Esta tecnologia introduziu várias novas tecnologias, inclusive gestão de estado, que não depende de *cookies* ou re-escrita de URL.

**RubyonRails[9]** é um *meta-framework* gratuito que promete aumentar velocidade e facilidade no desenvolvimento de sites orientados a base de dados (*database-driven web sites*), uma vez que é possível criar aplicações com base em estruturas pré-definidas. Frequentemente referenciado como *Rails* ou *RoR*, o *RubyonRails* é um projecto de código aberto escrito na linguagem de programação *Ruby*. As aplicações criadas utilizando o *framework Rails* são desenvolvidas com base no padrão de projecto *MVC* (*Model-View-Controller*).

**XML/XSL[10]** é um subtipo de *SGML* (*Standard Generalized Markup Language* – Linguagem Padronizada de Marcação Genérica) capaz de descrever diversos tipos de dados. É independente do *software* e do *hardware*, tendo-se tornado o formato mais usado da *Internet*. O XSL é a linguagem que define, transforma e formata o documento XML.

**MySQL[11]** é um Sistema de Gestão de Base de Dados (SGBD), que utiliza a linguagem SQL (*Structured Query Language*) como interface. É actualmente um dos SGBD mais utilizados.

**Oracle[12]** é um SGBD muito robusto e exige bastante do *hardware* para uma boa performance. A *Oracle* também criou a linguagem de programação PL/SQL, utilizada no processamento de transacções. Normalmente é escolhida e utilizada por grandes empresas, por ser um SGBD que prima pela segurança, pois essa é uma das principais vantagens da *Oracle*.

**POSTGRES[13]**, é um sistema SGBD (sistema de gestão de base de dados), desenvolvido como projecto de código aberto.

## 4.2 Tecnologias do lado Cliente

**HTML[14]** é uma linguagem com a qual se definem as páginas *Web*, trata-se de um conjunto de *tags* que servem para definir a forma na qual se apresentará o texto e outros elementos na página. Este tipo de documento pode ser interpretado pelos browsers. Esta tecnologia derivou-se da junção dos padrões *HyTime* e *SGML*.

*Hytime* caracteriza-se por padrões que representam estruturas de hipermédia e conteúdo baseado em tempo. Um documento é visto como um conjunto de eventos concorrentes dependentes de tempo (áudio, vídeo, etc.), conectados por hiper-ligações.

*SGML* é um padrão de formatação de textos. Não foi desenvolvido para hipertexto, mas tornou-se conveniente para transformar documentos em hiper-objectos e para descrever as ligações.

É a linguagem de *Web Design* mais fácil de usar e a mais popular. Esta linguagem é tão simples que pode-se digitar a sintaxe num editor de texto, guardar o ficheiro com extensão “.HTML” e imediatamente ter uma página *Web*.

**JavaScript[15]** é uma linguagem de programação *Web* que é executada do lado cliente, desenvolvida pela *Netscape*. Essa linguagem possibilita a adição de recursos dinâmicos às páginas HTML.

**CSS *CascadingStyleSheets* (CSS - Folhas de Estilo em Cascata)[16]**, tecnologia que permite criar páginas *web* sob determinada apresentação (formatação e visualização). Esta tecnologia permite fazer muitas coisas que não era possível utilizando somente HTML, como incluir margens, tipos de letra, fundos, cores, etc. CSS pode ser escrito dentro do código HTML da página web.

### 4.3 Outras técnicas utilizadas

**Jquery**, é uma *Framework* para ajudar os programadores a se concentrarem na lógica do sistema da *Web* e não nos problemas de incompatibilidade dos navegadores actuais. É uma biblioteca de *javascript* que simplifica muito o desenvolvimento de eventos, animações pois tem *plugins* para auxiliar no desenvolvimento do mesmo.

**Ajax, *AsynchronousJavaScript* and XML** não é uma tecnologia mas sim uma técnica de programação que possui a capacidade de retornar uma informação de forma dinâmica sem a necessidade de se recarregar toda a uma página e seu conteúdo.

Esta técnica recorre a várias tecnologias já citadas, principalmente *JavaScript*, utilizando a comunicação cliente servidor, e XML encapsulando a informação, cada uma a evoluir a sua maneira e a convergir de uma maneira poderosa, podemos ainda citar:

- Apresentação baseada em padrões, que utilizam XHTML e CSS.
- Exibição e interacção dinâmicas por meio de DOM (*DocumentObjectModel*).
- Troca e manipulação de dados por meio do uso de XML e XSLT.

- Recuperação assíncrona de dados com *XMLHttpRequest*.
- E *JavaScript*, que junta tudo.

A implementação mais comum do Ajax é baseada na classe *XMLHttpRequest*, que faz o *midlewaer* entre o servidor e o browser.

## 4.4 Ferramentas utilizadas

Neste capítulo serão apresentadas as ferramentas que estão a ser usadas para o desenvolvimento da plataforma:

- **Geany** – ferramenta de edição de texto.
- **PHPMyAdmin**– ferramenta de administração de Base de Dados.
- **FileZilla(Client)** – ferramenta FTP para transferência de dados.
- **Apache** – servidor http open source.

## 4.5 Tecnologias escolhidas para o desenvolvimento do SI e a razões

Para o desenvolvimento da plataforma foi escolhido as seguintes linguagens:

- PHP5.0,
- HTML,
- MySQL,
- Javascript,
- Ajax,
- JQuery,
- CSS.

### As razões da escolha daquelas tecnologias:

**HTML**, por se tratar do desenvolvimento de uma aplicação Web, e o HTML é uma linguagem que é usada para a definição das páginas Web.



**CSS**, por se tratar de um desenvolvimento de aplicação Web, e por ser uma tecnologia complementar ao HTML que permite tratar da formatação e visualização das páginas Web.

**JavaScript/Jquery**, é utilizada para evitar o problema de incompatibilidade entre *browsers*, pois cada *browser* tem um “*standard*” para o *JavaScript*.

**MySQL**, um dos factores da sua utilização é que o MySQL está disponível para praticamente todos os sistemas operativos (*Linux, Microsoft, Mac*), por ser um *software* “*open-source*”, também porque o serviço de hospedagem da aplicação, trabalha com o MySQL e a linguagem PHP. Para além dos factores mencionados acima o MySQL também foi escolhido por apresentar as seguintes características técnicas:

- Alta compatibilidade com linguagens como *PHP, Java, Python, C#, Ruby* e *C/C++*;
- Baixa exigência de processamento (em comparação como outros *SGBD* por exemplo a *Oracle*);
- Vários sistemas de armazenamento de dados, como *MyISAM, MySQLCluster, CSV, Marge, InnoDB*, entre outros;
- Recursos como *transaction* (transacções), ligação segura, indexação de campos de texto replicados;

**PHP**, as razões da sua escolha foram os seguintes:

- Velocidade de resposta;
- Portabilidade – independência de plataformas;
- Ser orientada a Objectos, utiliza a extensão PDO (*PHP Data object*);

O PDO apresenta uma interface rápida, segura, flexível e leve no acesso e manipulação dos dados da Base de Dados. Permite acima de tudo o controlo das transacções, suportando a inicialização e encerramento de transacções de forma padronizada com o *beginTransaction, commit e rollback* o que traz um ganho significativo na velocidade de produção do projecto. Utiliza o *exception* (classe *PDOException*) para manipulação dos erros e outras inconformidades, permitindo formatar as saídas de erro e verificar de forma agradável a origem dos mesmos usando o *catch(PDOException \$e)*.

- Sintaxe similar a Linguagem C/C++;

- Segurança (protecção contra *SQL injection*), por utilizar o PDO, instruções de *regex* ou *mysql\_escape\_string()*, *stripslashes()* (regaste de dados da Base de Dados) e *addslashes()* (inserção dos dados na Base de Dados) não permitem resolver o problema do *injection* de todo. O PDO veio introduzir instruções *preparedstatements* e filtragem de campos com *bindParam* para esse fim com maior eficácia (ganho na extensão de código), eficiência e rapidez.
- Tecnologia estável, flexível e consome menos recursos de *hardware*;
- Os custos associados à tecnologia PHP são relativamente inferiores, em relação às outras tecnologias, na medida em que possui licença gratuita e é uma tecnologia *open-source*.

#### 4.5.1 Problemas a resolver

As tecnologias escolhidas irão resolver um conjunto de problemas que a ANET-PP apresenta. Esses problemas serão descritos a seguir:

**Problema 1** - O SI não possuía o mesmo comportamento nos browsers mais utilizados (*Chrome, Firefox, IE 7, IE 8, Opera, Safari...*), uma das discrepâncias residia na abertura dos submenus: existia a necessidade de sair a aplicação e efectuar o login novamente.

**Problema 2** - Os dados na BD apresentavam redundância. Problema criado pela necessidade de serem desenvolvidas novas funcionalidades em tempo *record*, uma vez que a ANET não poderia ficar dependente do tempo que aqueles novos desenvolvimentos levariam a ser realizados do ponto de vista da eficiência. A sensibilidade de alguns utilizadores no preenchimento de determinados formulários, em que para o mesmo tipo de informação, inserida em diferentes momentos de tempo, existiam diferenças ao nível dos caracteres que compunham essa mesma informação o que gerava mais um registo na Base de Dados com dados em tudo semelhantes e redundantes.

#### 4.5.2 Melhorias aos problemas:

- **Problema 1:** remodelar e desenvolver uma nova plataforma implementada com recurso a tecnologias que possibilitassem uma maior eficiência nos desenvolvimentos: PHP5, JQUERY e Ajax, de forma a tornar funcional.

**-Problema 2:** foi necessário reaplicar a normalização na Base de Dados, com o intuito de remover a redundância e obter conformidade nos dados e consequentemente aplicação da reestruturação a algumas tabelas e a reestruturação de funcionalidades da plataforma. Implementar validação (inserção obrigatória de caracteres específicos para o campo) nos campos críticos de formulários e verificação (verificar se os dados a pesquisar existem na Base de Dados) em campos de pesquisa dos quais estejam dependentes operações de risco (como pagamento de pedidos de competências, associação de documentos ao membro respectivo).

## 5 Planeamento e Gestão do Projecto

Esta secção inclui a descrição de todo o planeamento e gestão do projecto feito ao longo do estágio, focando a metodologia utilizada para o fazer, a atribuição de responsabilidades mediante a metodologia seleccionada e os seus artefactos. É feita uma breve descrição de como foi elaborado o plano, execução e controlo das tarefas que foram desenvolvidas até ao término do projecto. Por fim é feita uma breve descrição da arquitectura da aplicação.

### 5.1 Metodologia usada para o desenvolvimento do Projecto

O projecto foi desenvolvido apoiando-se na metodologia SCRUM [17] que é um processo de desenvolvimento iterativo e incremental para a gestão de projectos e desenvolvimento ágil de software. Iterativo porque o trabalho é dividido em iterações (*Sprint*), normalmente com duração entre 2 a 4 semanas.

Abaixo é apresentada a figura da metodologia seguida de explicação da mesma:

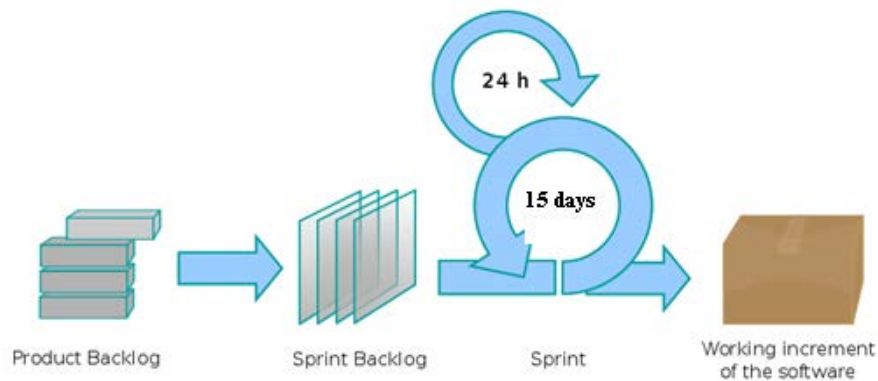


Figura 2- Metodologia SCRUM

#### 5.1.1 Fases do SCRUM

A primeira fase:

- **Planning:** definição do *BackLog* inicial da equipa que irá trabalhar no projecto, identifica-se os riscos e como controlá-los, verifica-se a infraestrutura para o desenvolvimento, as ferramentas necessárias, os custos e ainda outras actividades necessárias antes de se começar o projecto.
- **Architecture/HighLevel Design:** Projecta-se como os itens do *ProductBacklog* foram implementados. Revisam-se cada item e identificam-se as mudanças necessárias na arquitectura já existente do sistema ou redesenha-se a arquitectura em caso de novo produto. Deve ser uma arquitectura de alto nível, sem muitos detalhes já que o produto será melhor conhecido a medida que os *Sprints* forem ocorrendo.

#### A segunda fase:

- **Development:** É um ciclo iterativo de desenvolvimento, o *Sprint*, com um período pré-definido, normalmente entre 2 a 4 semanas, onde ocorrem as actividades necessárias para o desenvolvimento do produto. As iterações ocorrem quantas vezes forem necessárias até que o produto fique pronto e então ocorra a fase de *Closure*.

#### A terceira fase:

Possui apenas a fase **Closure**, e ocorre ao final da segunda fase, quando a equipa considera que o produto está pronto o suficiente para um *release*.

**Closure:** Inclui a preparação para um *release*, como um teste integrado do sistema, a documentação final, testes de usabilidade e outras actividades necessárias para concluir a fim do projecto.

### 5.1.2 Artefactos do SCRUM

Nessa seção será descrita os principais artefactos que definem o SCRUM do projecto desenvolvido:

**ProductBacklog**, que é a lista de funcionalidades com as respectivas prioridades definidas pelo *ProductOwner*. Sendo um projecto de remodelação e desenvolvimento de um Sistema de Informação já existente o *ProductOwner*, teve a oportunidade de estabelecer todas as funcionalidades, com as respectivas prioridades a serem desenvolvidas durante o período de estágio.

**Sprint:** Após listar todas as funcionalidades necessárias para o término do projecto, foram agrupados em incrementos que não devem ter duração superior a 30 dias de desenvolvimento. Cada um dos *sprints* teve uma semana de duração.

Para cada semana foi feito um relatório com o desenvolvimento das tarefas, esses relatórios são gerados pela ferramenta “*Sprintometer*”, conforme a figura abaixo:

General	Stories	Track Chart	Scope Chart	3D Burn Down	Resources & Budget	Story Readiness Report	Summary Report	Workload Report		
						Estimation\Date:	Apr 18	Apr 19	Apr 20	Apr 21
						Done %:	30% (15)	50% (25)	80% (40)	90% (45)
						Coded %:	30% (15)	50% (25)	80% (40)	90% (45)
						Tested %:	n/a	n/a	n/a	n/a
						Done today/to do:	15/15	10/10	15/5	5/5
						Coded today/to do:	15/15	10/10	15/5	5/5
Story ID, Task#	Story Name, Task Name		Assigned 1	Assigned 2	Tested today/to do:	0/0	0/0	0/0	0/0	0/0
10	Módulo Documentos				Done %:	30% (15)	50% (25)	80% (40)	90% (45)	
✓ 1	Reestruturação das tabelas		Jennifer Santos		10	✓ 10/0	✓ 0/0	✓ 0/0	✓ 0/0	
✓ 2	Alteração e reestruturação do código.		Jennifer Santos		15	5/5	✓ 10/0	✓ 0/0	✓ 0/0	
✓ 3	Entrega de documentos		Jennifer Santos		10	0/10	0/10	✓ 10/0	✓ 0/0	
4	Entrada de documentos em separado		Jennifer Santos		15			5/5	5/5	

Figura 3 - Sprint Semanais

Este tipo de relatório é muito interessante para a (orientadora/cliente), pois permite ver quais as tarefas que já estão completas e qual o andamento das que estão em execução.

Como podemos ver, temos um posicionamento consolidado da evolução de cada tarefa. Permite-nos saber qual o recurso que pode ser entregue ao cliente, quais aqueles que estão com problemas ou que tiveram pedidos de alteração como é o caso da tarefa 4 (que ficou pendente) e ainda se encontra em desenvolvimento.

Esta tabela é um exemplo de uma das semanas realizadas pela estagiária. Os restantes *sprints* semanais encontram-se descritos no documento “Anexo F”.

**SprintBacklog:** É uma lista que contém as tarefas que o *ScrumTeam* irá trabalhar durante o *Sprint*. Os itens do *ProductBacklog* com a lista de prioridades definidas pelo *ProductOwner* (a orientadora), em que aquelas tarefas devem ser cumpridas dentro do *Sprint*, caso exista incumprimento das tarefas, estas devem fazer parte do próximo *SprintBacklog*.

Neste projecto os módulos desenvolvidos sujeitos a definição dos *sprintBacklog*, são:

- Módulo Membros, com uma estimativa de três semanas.

- Módulo Inscrições/Estágios, com uma estimativa de três semanas.
- Módulo Documentos, com uma estimativa de duas semanas. Deste módulo será implementado só o submenu “Gestão de Documentos”.
- Módulo Declarações, com uma estimativa inicial de cinco semanas, acabou por não ser desenvolvido pela estagiária pela escassez de tempo para realizar os devidos testes à aplicação.

Para cada um destes módulos foi gerado um gráfico com a evolução correspondente para o projecto, informação esta que consta do documento “Anexo F”, entregue semanalmente ao *ScrumMaster*.

**BurndownChart:** Gráfico que acompanha a evolução do *Sprint*, criado a partir da lista de tarefas do *Sprint*, que vai sendo actualizada pela estagiária a medida que for concluído as tarefas pendentes. Com este gráfico o *ScrumMaster* poderá acompanhar a evolução do projecto e a própria estagiária terá uma noção do número de horas restantes para a conclusão de cada tarefa.

Os gráficos serão desenvolvidos com a ajuda da ferramenta *Sprintometer*[18] e anexadas ao documento. Abaixo serão apresentados alguns gráficos desenvolvidos com a ferramenta:

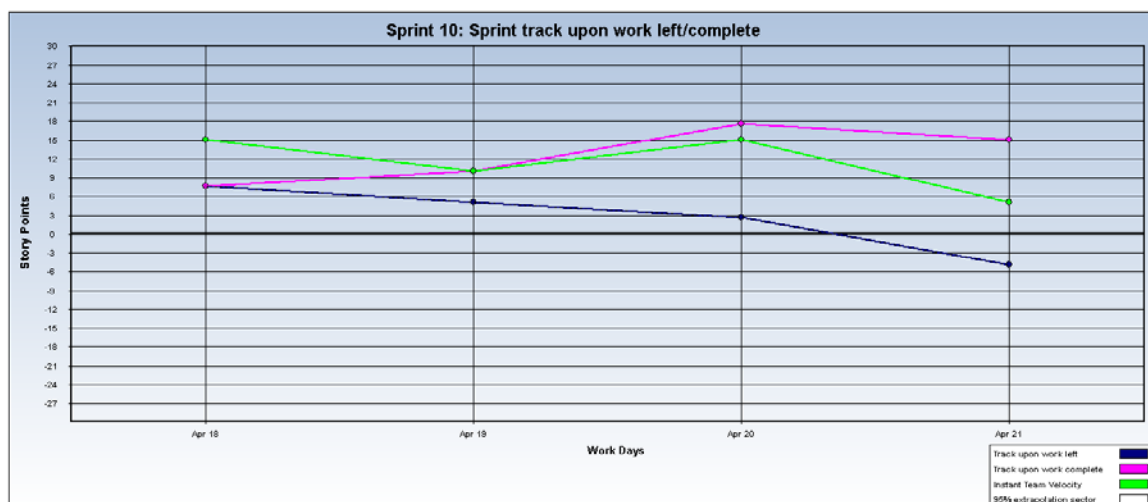


Figura 4 - TrackChart

Neste gráfico, a linha verde indica a velocidade da equipa.

Ao somarmos a velocidade de todos os dias e dividirmos pela quantidade de dias, encontramos a velocidade média da equipa. Com isso, podemos verificar quantas tarefas faltam para ser concluído e avaliar se estamos próximos ou não de entregar as funcionalidades atribuídas no prazo definido no início do *sprint*.

No gráfico *BurnDown* que é gerado pela ferramenta “*Sprintometer*”, encontra-se este tipo de previsão de término das funcionalidades (tarefas), baseando-se nessa velocidade média da equipa. Abaixo será apresentado o gráfico da semana que tem sido de exemplo, 18 a 21 de Abril:

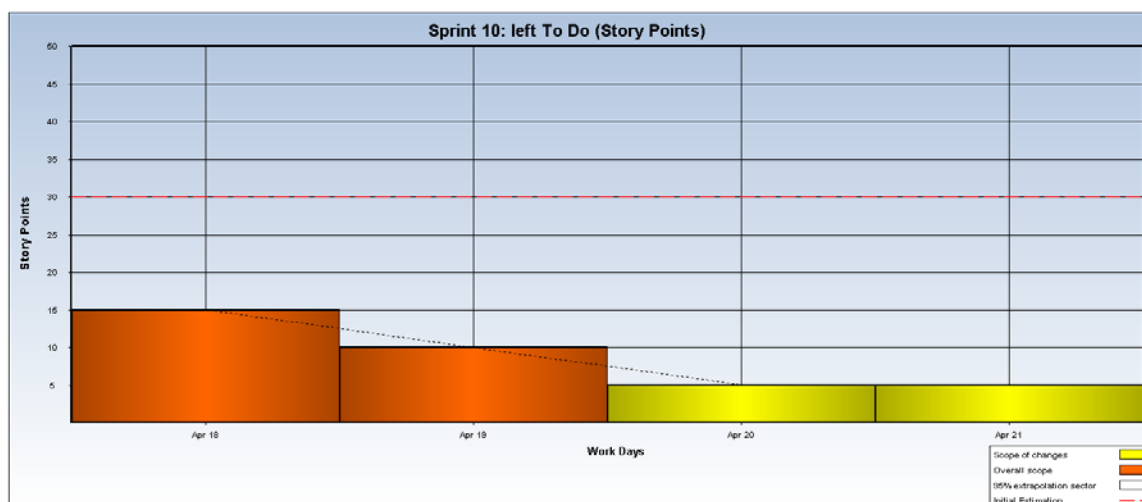


Figura 5 - BurndownChart

Este gráfico permite ter uma visão da evolução diária do projecto, além de nos permitir fazer projecções do término do *sprint*, baseado na velocidade da equipa.

A ferramenta *Sprintometer* gera este gráfico automaticamente, baseado nas informações de evolução (s.p concluídos / s.p restantes) que inserimos nas tarefas.

Com esses gráficos podemos verificar se houve algum problema, por exemplo, se temos uma média de conclusão 20 pontos por dia durante 6 dias do *sprint* e no sétimo dia foi feito apenas 13, isso indica que alguma coisa fora do planeado ocorreu, por exemplo uma mudança inesperada, ausência da estagiária, etc... É importante realizar-se o cruzamento deste tipo de informações para que (orientadora e a estagiária) não se perca o controlo do andamento do projecto, verificando que a estagiária tem uma produção média X e também ficará a saber que por algum motivo a estagiária teve problemas no seu desempenho.



Para o exemplo da semana 18 a 21, as conclusões tiradas a partir do gráfico “*Burndown*” acima, nos mostra que a estimativa inicial do *sprint* era de 30 horas e que após o primeiro dia restam 15 pontos para serem entregues. Após o segundo dia, restaram 10 pontos, no terceiro 5 pontos, sendo que 5 pontos foram de solicitações de mudanças (parte amarela da barra), para o término das funcionalidades.

## 5.2 Equipa do projecto

Nessa seção será descrita os papéis e as responsabilidades no desenvolvimento do projecto seguindo a metodologia SCRUM.

O SCRUM define basicamente 3 papéis, o *ProductOwner*, *ScrumMaster* e *ScrumTeam*.

### 5.2.1 ProductOwner e ScrumMaster

Neste projecto a orientadora vais desempenhar o papel de *ProductOwner* e de *ScrumMaster*.

De *productOwner* por ser a representante do cliente, a que vai representar os interesses de todos os envolvidos com o software. É quem vai definir a lista de prioridades das funcionalidades que compõem o *ProductBacklog*.

Também vai desempenhar o papel de *ScrumMaster*, para além de representar o cliente será a gestora do projecto, tendo a responsabilidade de garantir o trabalho da estagiária através da remoção de obstáculos e dificuldades identificadas durante a execução das tarefas.

### 5.2.2 ScrumTeam

Neste projecto a equipa responsável pelo desenvolvimento do produto de acordo com a lista de prioridades definidas pelo *ProductOwner*, vai ser da responsabilidade da estagiária.

## 5.3 Plano, execução e controle do Sprint

Os *Sprint* são definidos semanalmente, em que o plano de execução e controlo do *Sprint* passa por três fases:

No início de cada *Sprint* ocorre uma reunião com o *ProductOwner/ScrumMaster*, para serem definidos as tarefas com mais prioridades.

De seguida é feita uma reunião no meio da semana com o *ProductOwner/ScrumMaster*, onde será este é informado sobre a evolução do projecto apresentado igualmente os gráficos “*BurndownChart*”. A orientadora poderá verificar as barreiras com que a estagiária se deparou no desenvolvimento, podendo assim orientar e eliminar essas barreiras.

No final de cada *Sprint*, é feita uma reunião informal onde a estagiária apresentada um documento descrevendo o que foi implementado acompanhado de ilustrações como as figuras acima apresentadas (Figura 3,4 e 5) onde se mostra o que foi alcançado no respectivo *sprint*.

Finalmente realizam-se os testes de funcionalidades nas tarefas implementadas de forma a verificar o que funciona correctamente, o que pode ser melhorado, que acções serão tomadas para melhorar e quais as funcionalidades que não foram implementadas e que passaram a fazer parte do novo *sprint*.

## 6 Arquitectura da Aplicação

Nessa secção será apresentada os pontos relevantes da arquitectura da aplicação e a estrutura da mesma de forma a perceber melhor a arquitectura organizacional do mesmo, para mais detalhes ver o “Anexo B”.

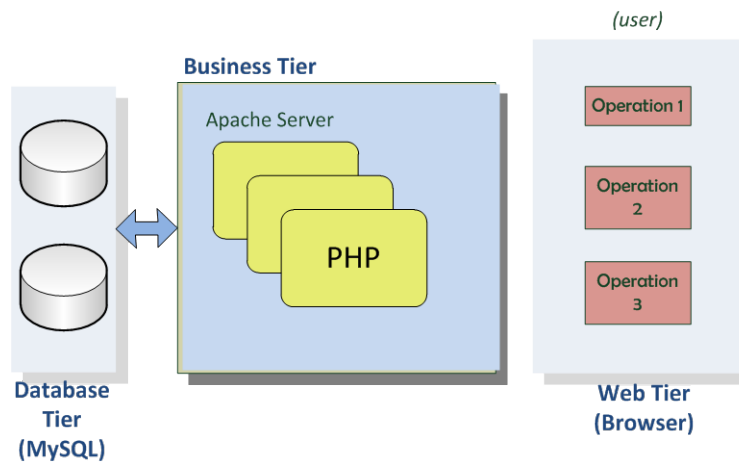


Figura 6 - Arquitectura da aplicação

### 6.1 Camada de Apresentação ou Interface

Esta é a camada através da qual o utilizador interage com a aplicação. Sendo esta uma aplicação *Web Based*, o seu ponto de interface será o browser. Nesta camada encontram-se os ficheiros *PHP* contendo *HTML*, *JavaScript* e *CSS* que implementam a interface *Web*. Aqui é processada o input e o output de dados, com recurso aos serviços oferecidos pela camada de negócio.

### 6.2 Camada de Negócio

Esta corresponde à camada de aplicação onde são implementadas as regras de negócio através de módulos de acesso às bases de dados escritos em *PHP*. Os dados provenientes da camada de dados são manipulados de acordo com as necessidades e lógica de negócio da empresa entregando-os à camada de apresentação.

## 6.3 Camada de Dados

Nesta camada encontram-se todas as estruturas de armazenamento de dados onde é guardada toda a informação que a aplicação apresenta. É a camada responsável pela manutenção, integridade dos dados e informações da aplicação.

## 6.4 Estrutura geral da aplicação




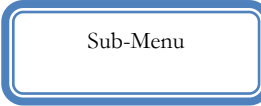

Nível	Descrição	Exemplo
 Aplicação	A aplicação é composta por vários grupos de módulos	S.I -ANET'
 Grupo de Módulos	Um grupo de módulo separa uma aplicação em interface com requisitos lógicos	Interface com o utilizador
 Módulos	Um módulo é um conjunto de sub-menus do mesmo domínio mas com conteúdos e estrutura diferentes	Membro
 Sub-Menu	Sub-menu é um grupo de operações pertencentes a um domínio comum	Planear Formações
 Operações	Operações são uma acção que o sistema executa, uma acção está associada a uma edição, remoção, visualização	Adicionar Formações

Tabela 2 - Estrutura geral da aplicação

### 6.4.1 Aplicação ANET'

A aplicação ANET' será composta por dois grupos: Aplicação ANET' que será a composição dos vários módulos e a parte da conexão com o Servidor Apache

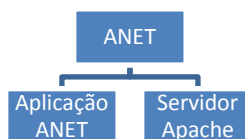


Figura 7 - Decomposição do sistema ANET em módulos.

### 6.4.2 Módulos da Aplicação ANET

A aplicação ANET encontra-se devida em 7 módulos, sendo elas representadas na figura a seguir:

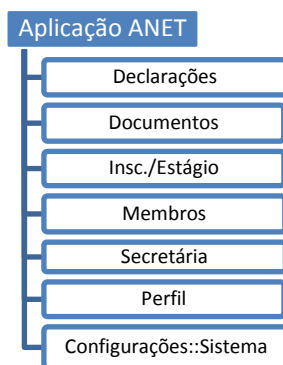


Figura 8 - Decomposição da aplicação ANET em módulos.

**Nota:** Estes são os módulos que foram desenvolvidos pela estagiária para nova plataforma, sendo que a aplicação irá conter outros módulos que serão posteriormente desenvolvidos.

Módulo secretaria e declarações só encontram desenvolvidos os *layouts*.

Módulo Perfil e Configurações::Sistema foram desenvolvidas pela orientadora.

### 6.4.3 Estrutura do código desenvolvida para Aplicação:

A aplicação encontra-se sub-dividida em pastas, dispostas segundo uma ordem lógica do ponto de vista estrutural, que agrupam um conjunto de ficheiros (.php, .css ou .js) de acordo com a sua finalidade.

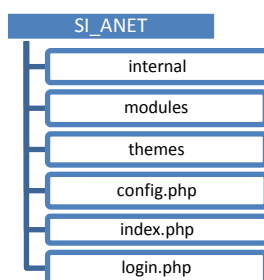


Figura 9 - Estrutura da Aplicação.

## 7 Testes

Ao longo da fase de implementação a aplicação desenvolvida foi submetida a vários testes, com o intuito de verificar erros ou falhas que fossem surgindo no desenvolvimento de cada módulo, e de verificar a integração entre os vários módulos tendo em conta o objectivo acordado. Foram igualmente realizadas rectificações nos erros encontradas após os testes.

Seguidamente, surge uma descrição dos vários tipos de testes efectuados.

### 7.1 Testes funcionais

Este tipo de testes tem a finalidade de assegurar a boa qualidade da aplicação, mediante verificação dos requisitos implementados, de forma a detectar o maior número de incongruências para que fossem rectificadas atempadamente. A realização deste tipo de teste implica contemplar:

- **Testes de Ambiente:** relacionados com o aspecto da Aplicação;
- **Testes de Navegabilidade:** avaliação da navegação entre Funcionalidades de cada módulo do SI, isto é verificar se todas as navegabilidades (botões, links, etc.) existentes na aplicação funcionam correctamente e se apresentam o resultado pretendido.
- **Teste Portabilidade:** este tipo de teste tem como finalidade verificar se a aplicação *Web* funciona correctamente em tendo em conta determinados cenários: *browsers* (mais usados), sistemas operativos e resoluções de ecrã.
- **Testes de Integridade:** testes realizados para validar os dados inseridos pelo utilizador e consequentemente avaliar a robustez do objectivo de estudo (resistência a falhas). A realização deste tipo de teste implica garantir:

- Impossibilidade de acesso à aplicação sem que se tenha efectuado o login: simular a autenticação de um utilizador não autorizado e verificar o comportamento.

- Conceito de *transacção*, verificando o cumprimento de quatro propriedades que o caracterizam (propriedades ACID): **atomicidade** (cada transacção que contenha várias transacções deverá ser executada na íntegra e os seus resultados efectivados, em caso de ocorrência de falha, nenhum resultado obtido até ao momento da falha será efectivo); **consistência** (relativo à concorrência entre transacções, ou seja, se num mesmo processamento existir concorrência entre transacções, o resultado deverá ser o mesmo se aquelas transacções fossem executadas de forma sequencial); **independência** (os resultados da transacção apenas serão utilizados por aquela transacção até ao seu término e efectivação de resultados) e **durabilidade** (pretende-se que garanta que nenhum dado seja perdido durante a transacção e que o resultado seja efectivo).

A busca pelo desenvolvimento de produtos de *software* com uma maior qualidade tem aumentado a importância da etapa de testes dentro do processo de desenvolvimento. Esta qualidade pode ser obtida com a utilização de metodologias de desenvolvimento e ciclos de testes bem definidos. A aplicação de testes de *software* de forma sistemática e automatizada consiste num meio para se alcançar qualidade no processo de *software* com um menor esforço e custo.

Através dos testes funcionais realizados e dos resultados obtidos com a inserção de diferentes valores, na sequência do término de cada módulo, foi possível verificar falhas e proceder às devidas correcções atempadamente, obtendo-se uma melhoria significativa no processo de desenvolvimento da aplicação. A adopção desta metodologia proporcionou diversos benefícios relacionados com um menor esforço gasto na manutenção e com a automação na execução de Testes Funcionais, não permitindo que esta tarefa fosse realizada na fase final do projecto o que redobraría os esforços.

## 7.2 Teste de Escalabilidade

Tipo de teste de desempenho que compara o desempenho da aplicação e uma carga de trabalho de referência conhecida. Este tipo de teste também é usado para validar e avaliar a aceitabilidade dos limites operacionais do sistema de acordo com cargas de trabalho variáveis.

Ferramenta a utilizar:

Para efectuar tais testes foi usada a ferramenta “*httperf*”, responsável pela medição do desempenho de servidor (*throughput*) e funciona da seguinte forma: o cliente gera carga para o servidor e a avaliação ocorre mediante o tempo de resposta.

Os dados obtidos encontram-se descritos no quadro abaixo:

	Número Ligações ( <i>Connections</i> )	Pedidos ( <i>Request</i> )	Resposta ( <i>Replies</i> )	Tempo ( <i>Timing</i> )	Erros ( <i>Errors/Failures</i> )
Teste 1	250	500	500	25.053 s	Nenhum
Teste 2	500	1000	1000	50.087 s	Nenhum
Teste 3	1000	2000	2000	100.016 s	Nenhum
Teste 4	1500	3000	3000	150.017 s	Nenhum

Tabela 3 - Resultados de testes de escalabilidade

### 7.3 Testes de Performance

Páginas leves e rápidas no carregamento potencia uma melhoria considerável do ponto de vista da performance, uma vez que quanto mais rápido o processamento mais rápido o tempo de resposta ao utilizador que consequentemente incrementa o grau de satisfação daqueles. Para tal, durante a implementação existiu a preocupação de se seguir determinadas regras que pudessem garantir uma boa performance.

Com o auxílio da ferramenta “*YSlow*”, que avalia a página segundo um conjunto de regras atribuindo uma classificação de A à F, já referidas anteriormente (ver secção 3.2.3 *Requisitos Não funcionais – Performance*).

A aplicação ANET- PDE apresentava, então uma classificação **C**, com limitações nas seguintes regras:



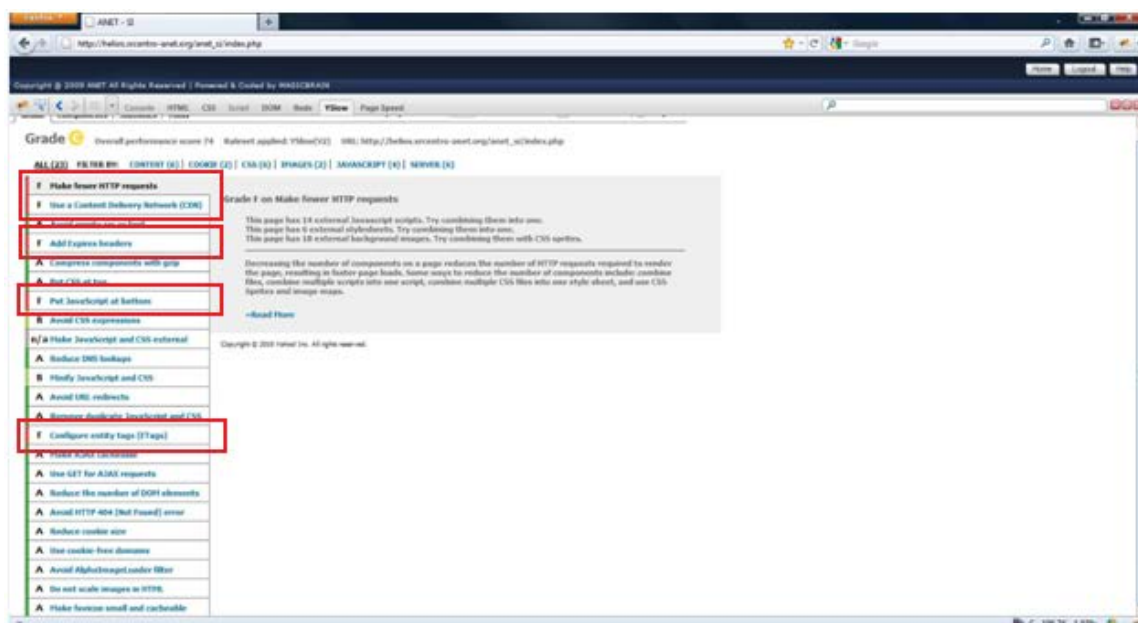


Ilustração 3 - Classificação ANET-PDE antes das melhorias

Com as melhorias realizadas, (referidas igualmente naquela secção), conseguiu-se obter uma classificação **B**, que resultou numa melhoria significativa do seu desempenho:

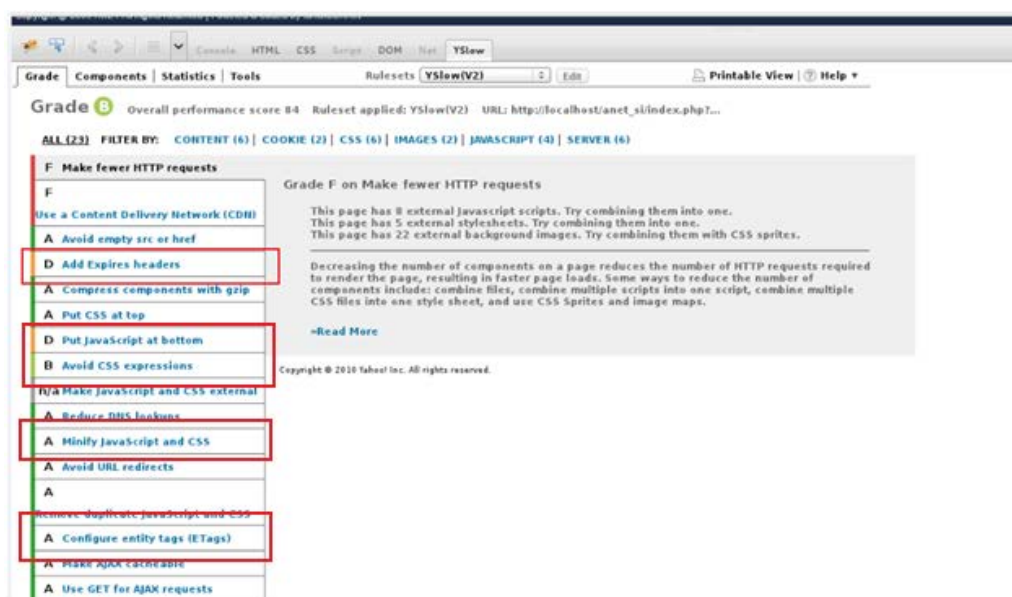


Ilustração 4 - Classificação da ANET-PDE depois das melhorias

## 7.4 Teste de Segurança

Este tipo de teste tem como finalidade verificar a segurança da aplicação, incluindo o acesso aos dados e acesso remoto a aplicação. Para tal, garantiu-se:

- A impossibilidade de acesso sem autenticação.

- A impossibilidade de injeção no código. O teste de segurança ao nível do “SQL-injection” foi feito seguindo alguns pontos do documento do “PHP *ApplicationSecurityChecklist*”[19]. Sendo listados a seguir os testes realizados:

**Basic:**

- Passwords guardadas de forma segura, utilizando encriptação recorrendo ao algoritmo SHA1.

**Input:**

- \$\_SERVER['PHP\_SELF'], está sujeito a vulnerabilidades, pelo que se recomenda evitar o seu uso ou utilizar o *escaped*, ou seja antecedido de barra invertida “\” para evitar leitura da variável; Na aplicação desenvolvida pela estagiária, foi evitado usar esta referência para se chegar ao caminho do ficheiro em execução.
- Validação dos Dados de Entrada.
- \0 (null) Descartado à entrada.
- Limitação do tamanho dos campos de input.
- Dados produzidos são “limpos”.
- URLs são “limpos” e protocolos desconhecidos ou não desejados, não são permitidos.

**File upload:**

- Passwords guardadas de forma segura.

**Database:**

- Dados inseridos nas bases de dados foram *escaped* (antecedidos de barra invertida “\”) ou utilizadas *statements* devidamente adequadas preparadas (*prepared statements*).
- *addslashes()* não é utilizado.

**Serving Files:**

- Directorias transversais não são permitidas (*directory-traversal*).

### Authetication:

- CAPTCHA é usado.

Abaixo serão apresentados alguns dos testes efectuados apenas para o módulo “Membros” e alguns dos erros que a estagiária se deparou ao realizar testes de segurança contra “XSS-Injection”:

ID Teste	Nome Teste			
		Resultados Obtido	Comportamento	Observações
1	Inserir <script>alert('xss');</script> no formulário de autenticação	Mensagem de erro: Password ou Utilizador errado!!	Funcional	
2	Inserir script no formulário de pesquisa rápida do submenu “Listagem membros” opção com foto.	Apresenta a tabela resultante sem nenhum valor.	Funcional	
3	Inserir script no formulário de pesquisa rápida do submenu “Listagem membros” opção sem foto.	Apresenta a tabela resultante sem nenhum valor.	Funcional	
4	Inserir scripts nos campos de pesquisa directa.	Mensagem de erro: Resultados: 0. Não possui resultados para a pesquisa efectuada. Escolhe o método de pesquisa correcta.	Funcional	
5	Inserir <i>scripts</i> no campo de inserção do número do membro para realizar pesquisa composta.	Resultado conforme o esperado, quando se tenta inserir valores não numéricos, o cursor fica parado a espera de valores do mesmo tipo.	Funcional	
6	Inserir <i>scripts</i> nos campos de classificação final e no campo à classificação final pesquisa composta.	Ao tentar inserir um valor que não seja numérico, o cursor fica parado a espera de ser inserido valores do mesmo tipo.	Funcional	
7	Inserir <i>script</i> no campo nome, no formulário de alterar dados do submenu Gestão de Formação.	Edição do dado com nome e retira as <i>tags</i> , por exemplo após, inserir: Alexandre Manuel <script>alert('xss');</script>, Alexandre Manel alert('\xss\');	Funcional	Este campo teve de ser protegido contra esse tipo de injectio, de modo a ficar funcional

8	Inserir <i>script</i> no campo telefone, no formulário de alterar dados do submenu Gestão de Formação.	O campo fica sublinhado, como aviso para que o utilizador insira números. ou "+", por ser um número de telefone.	Funcional	
9	Inserir <i>script</i> : <script>alert('xss');</script> no campo e-mail formulário de alterar dados do submenu Gestão de Formação.	Retirar os "<script>", por forma a evitar o bloqueio na exibição dos dados.	Funcional	
10	Inserir script no formulário para adicionar curso de formação.	Ao inserir esse script nos campos do formulário é retirado a <i>tags</i> de forma a não bloquear a listagem das formações após carregar no botão de voltar.	Funcional	
11	Inserir script no formulário de pesquisa de gestão de competências.	O cursor fica bloqueado a espera que seja inserido um valor numérico no campo de pesquisa.	Funcional	

Tabela 4 - Resultado de testes de segurança

Alguns erros encontrados e solucionados:

Erro	Módulo	Sequencia	Possível Solução
Bloqueio dos dados resultantes na lista de membros pertencentes a uma formação.	Módulo membro, submenu "Gestão de Formação", visualizar dados da formação	Sequência de introdução do <i>script</i> <script>alert('xss');</script> no campo no do formulário de pesquisa.	strip_tags(\$_POST['nome'])
No campo de pesquisa composta, erro ao inserir aqueles <i>script</i> .	Módulo membro, submenu "Pesquisa", opção de pesquisa composta	Na sequência de inserir o script <script>alert('xss')</script> no campo localidade.	strip_tags(\$_POST['nome'])

Tabela 5 - Erros encontrados/melhorados

Os restantes testes encontram-se documentados no “Anexo H”.

## 7.5 Após execução de testes

Nesta secção são apresentadas as correcções realizadas sobre pequenos erros e as melhorias aplicadas à plataforma mediante a submissão desta aos testes finais. Para cada tipo de teste efectuado será feita uma pequena descrição do que foi alterado ou melhorado.

Para os testes funcionais, houve a necessidade de alterar/melhorar algumas *queries* de pesquisa pois o tempo de resposta para algumas das funcionalidades, era demasiado longo face ao pretendido.

Procedeu-se à alteração da opção de ***adicionar competências*** uma vez que esta funcionalidade apresentava uma listagem com o tipo de competências e na mesma página, após pesquisa de membro, incluía as competências deste o que resultava num aumento de *scroll* realizado naquela página e tempos de carregamento da página. No sentido de evitar este inconveniente, dividiu os resultados por duas páginas para conseguir um melhor desempenho a nível do tempo de resposta.

A opção carregar documentos para determinado membro, os resultados apresentados não correspondiam aos requeridos para aquele membro pesquisado.

No que toca a portabilidade do *browser*, a estagiária alterou formatações de algumas imagens que estavam a criar alguns problemas para o *browser* IE, o que foi solucionado.

No documento de teste em anexo, são apresentados os testes e os problemas encontrados, devidamente solucionados.

### 7.5.1 Testes

Para os testes de performance, houve a necessidade de alterar alguns pontos entre os quais se destacam:

- Os códigos “*JavaScript*” e “*CSS*” foram passados para um ficheiro aparte, tendo sido mantidos códigos *javascript inline* específicos da página, no sentido da optimização.
- Foram realizadas configurações no “*httpd.conf*” a fim de possibilitar guardar ficheiros em cache, para que o carregamento das páginas se torne mais rápido: os ficheiros em cache impedem o carregamento contínuo de ficheiros. Foi definido uma data de expiração para os ficheiros que não sofrem alterações com muita frequência. Estas configurações encontram-se num ficheiro em “Anexo E”.

Face os testes de escalabilidade realizados, não foram verificadas quaisquer alterações a efectuar, uma vez que o número máximo de ligações em simultâneo será de 100, o que nesta aplicação não se traduz numa perda de potencialidades.

## 8 Plano de Trabalho

Neste capítulo serão apresentadas as tarefas atribuídas à estagiária e desempenhadas pela mesma durante o projecto de estágio.

A figura seguinte apresenta o *Diagrama de Gantt* elaborado inicialmente para o período de estágio, representando o escalonamento de tarefas previstas para aquele espaço de tempo.

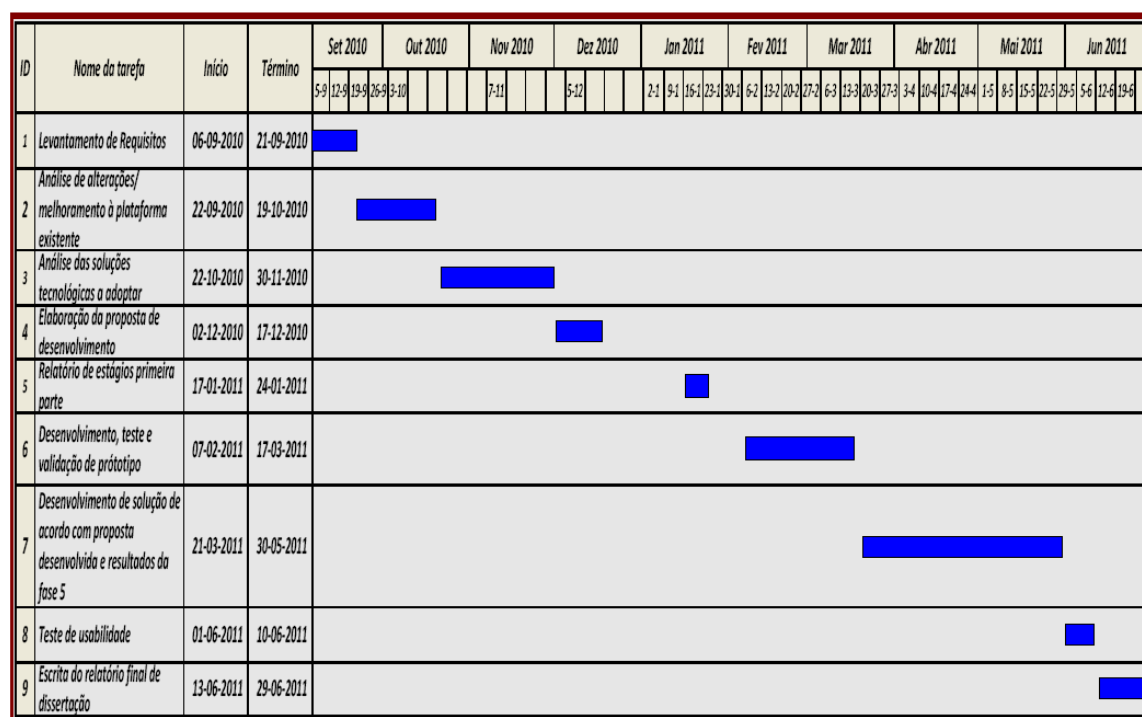


Figura 10 - Diagrama de Gantt inicial 1º Semestre – 10 Setembro

No entanto, aquela previsão sofreu algumas alterações. A ANET-PP possuía uma complexidade de raciocínio e extensão de módulos de tal ordem que existiu uma necessidade de repensar e subdividir as tarefas para que estagiária pudesse implementar, documentar e testar funcionalidades bloco a bloco com a finalidade de evitar a ocorrência dos problemas deparadas naquela plataforma.

A seguir será apresentado o *Diagrama de Gantt* elaborado para o segundo semestre já com os módulos a serem desenvolvidos:

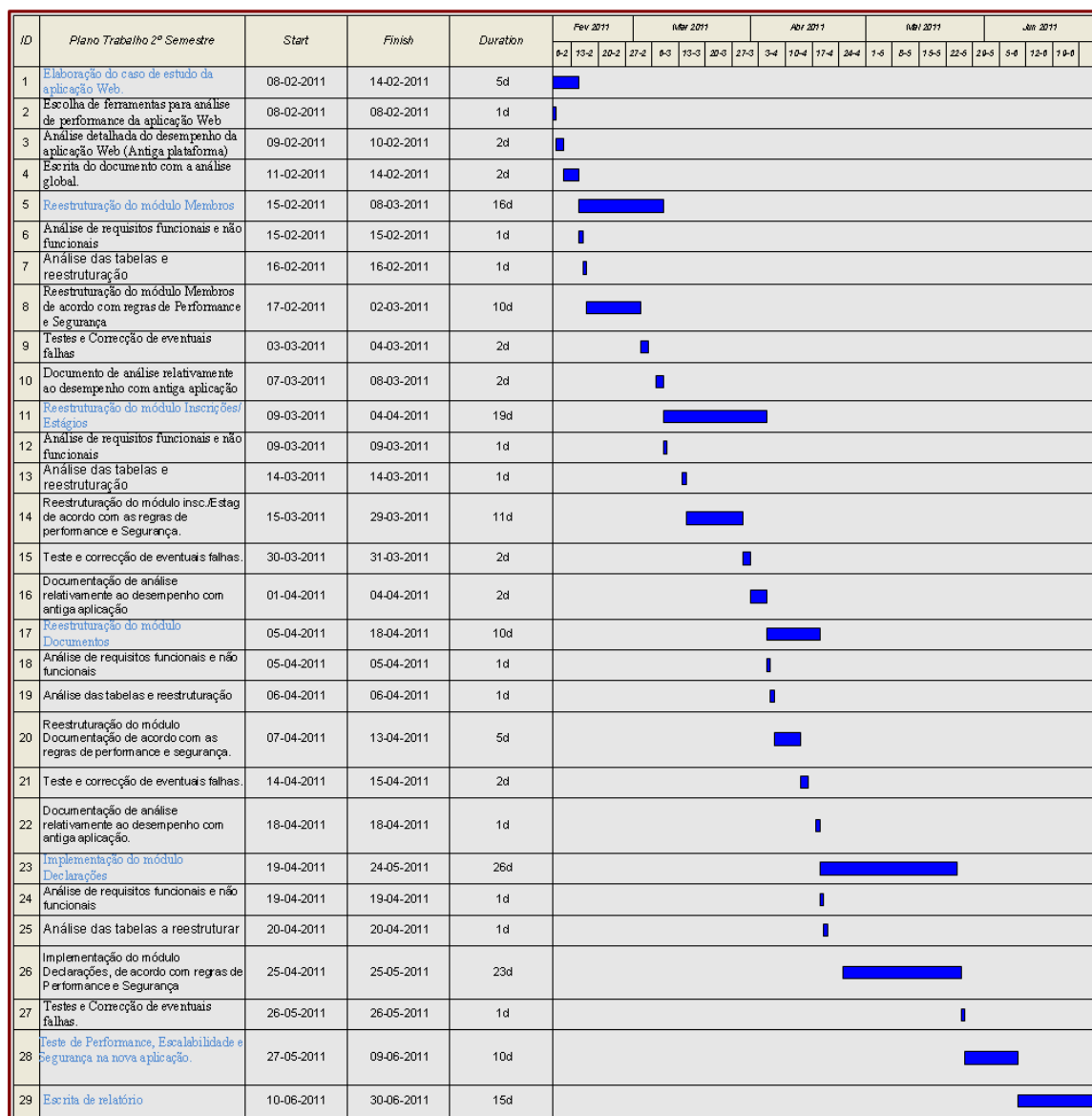


Figura 11 - Diagrama de Gantt inicial 2º Semestre – 7 Fevereiro

O novo *Diagrama de Gantt* com as actualizações finais, é apresentado em seguida.

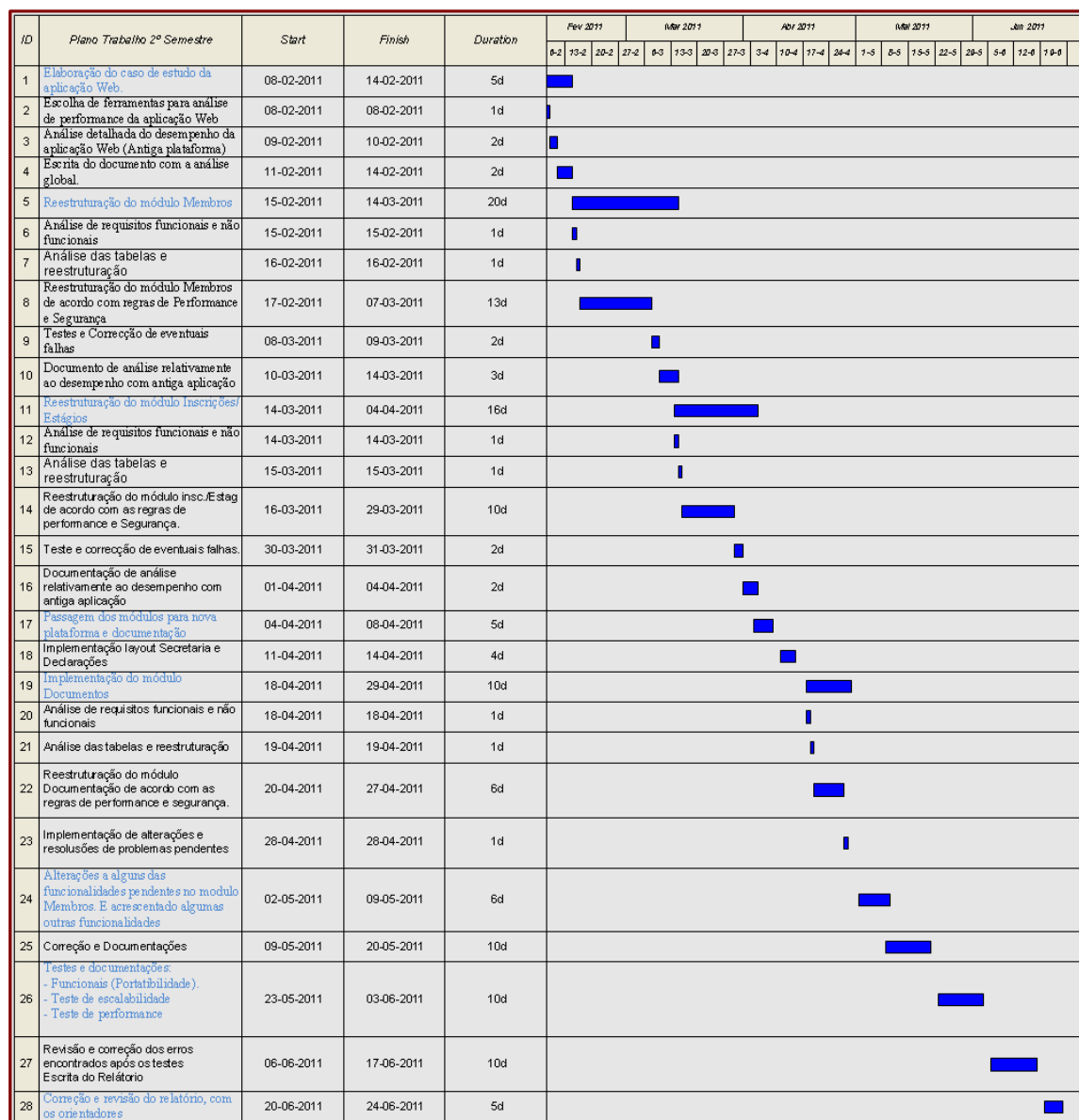


Figura 12–Diagrama de Gantt Final 2º semestre - Junho

As tarefas projectadas no diagrama da Figura anterior, foram concluídas com a excepção do módulo de **Declarações**, por questões de tempo que não deixava à estagiária margem de manobra para terminar o módulo de forma conveniente, isto é, realizar o seu desenvolvimento e incluir a fase de testes ao mesmo.



### **Fases do diagrama de Gantt:**

- **Elaboração de caso de estudo da aplicação ANET-PP**, fase para a escolha de ferramentas para a análise à aplicação, testes e elaboração da documentação referente à aplicação de forma a verificar os problemas desta e encontrar formas para implementar a ANET-PDE sem que aqueles problemas se verificassem.
- **Reestruturação/implementação dos módulos**, fase mais longa do escalonamento, referente à implementação dos vários módulos. Por cada implementação de módulo, foi feita uma reunião primária para verificar as funcionalidades a serem implementadas, analisar a Base de Dados e verificar as implicações associadas ao novo desenvolvimento. Cada implementação foi dividida em subtarefas com periodicidade de 2 semanas, de forma a existir um melhor controlo das mesmas e realização de **testes funcionais** mediante as implementações realizadas.

A secção **6. (Testes)** apresenta uma descrição deste tipo de testes realizado, a sua documentação encontra-se em anexo no ficheiro **teste\_funcionais.docx** (CD-ROM “Anexo G”).

- **Testes**, fase de elaboração de testes finais referentes a testes de performance, escalabilidade e segurança, com apresentação na secção **6. Testes** cuja especificação dos resultados dos mesmos se encontram documentados nos ficheiros anexados ao CD-ROM (Anexo H, I e J).  
Nesta fase também foram feitas refinações da aplicação ANET-PDE que consistiram em pequenas correcções de erros à medida que esta era submetida aos testes finais.
- **Escrita do Relatório**, fase para a elaboração do relatório de estágio e correcção de alguns detalhes nos vários documentos elaborados ao longo do mesmo.

## 9 Conclusões

Depois do projecto de estágio finalizado, é necessário tirar conclusões sobre o trabalho desenvolvido durante este período, e também sobre o futuro que este projecto terá.

Durante o desenvolvimento do projecto, a estagiária foi-se deparando com alguns desafios inerentes ao mesmo como: a tarefa de entender a antiga aplicação existente de modo a poder encontrar estratégias para o seu melhoramento, visto não se tratar de um projecto que foi começado de raiz; a adaptação às linguagens de programação; entender o modelo de implementação utilizado pela empresa para o desenvolvimento de aplicações Web; resolver problemas que foram surgindo com a implementação dos diversos módulos e também problemas na passagem dos módulos do modelo inicial (modelo de interacção com plataforma) para o modelo funcional.

Foram alcançados os objectivos estipulados para este estágio, uma vez que a estagiária conseguiu planear, desenvolver e analisar um projecto, utilizando as funcionalidades propostas. A documentação gerada consegue descrever o que se pretende do projecto e mostra o planeamento desenvolvido ao longo do mesmo, até chegar a um resultado final.

### Trabalho desenvolvido:

De início, o objectivo do projecto de estágio era analisar a estrutura da BD e implementar os módulos existentes da plataforma em produção na ANET com novas tecnologias. Mas dado o número elevado de módulos e a quantidade de informação na BD, foi reestruturada a planificação de estágio tendo sido implementados apenas alguns módulos, para que a estagiária tivesse a oportunidade de fazer uma análise mais profunda ao nível de performance, escalabilidade e segurança.

No final do estágio, os módulos projectados encontravam-se devidamente implementados e testados, tendo em conta que os parâmetros de análise ficarão prontos a serem usados. Posteriormente serão desenvolvidos os restantes módulos.

### Trabalhos Futuros:

O trabalho desenvolvido permite uma boa base de evolução para uma aplicação mais consistente e robusta, uma vez que nesta fase se encontram definidas as novas tecnologias a usar, a metodologia de desenvolvimento e os planos de testes que devem ser feitos à medida que se forem implementando os módulos.

Do lado do Modelo de Dados deverá continuar-se com o tratamento dos dados das tabelas relacionadas com os outros módulos a ser desenvolvidos: Secretaria, Declarações, Contabilidade, Documentação e Meu INETSYS.

## 10 Referências

[1]- [http://pt.wikipedia.org/wiki/Cross-site\\_scripting](http://pt.wikipedia.org/wiki/Cross-site_scripting)

[2]- [http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)

[3]- <http://developer.yahoo.com/yslow/>

SOUDERS, Steve. High Performance Web Sites - Essential Knowledge for Front-End Engineers, O'Reilly Media, September 2007

[4]-<http://explainextended.com/2010/05/27/left-join-is-null-vs-not-in-vs-not-exists-nullable-columns/>

-<http://www.getcon.com.br/blog-technoide/index.php/2010/04/sql-substituindo-subconsultas-por-juncoes/>

- <http://www.criarweb.com/artigos/subconsultas-em-sql.html>

[5] – <http://www.hpl.hp.com/research/linux/httpperf/docs.php>

[6]- [http://www.wordiq.com/definition/PHP\\_programming\\_language](http://www.wordiq.com/definition/PHP_programming_language)

-[http://www.php.net/manual/pt\\_BR/](http://www.php.net/manual/pt_BR/)

[7] - <http://pt.wikipedia.org/wiki/JSP>

[8] - <http://pt.wikipedia.org/wiki/ASP>

[9] - [http://pt.wikipedia.org/wiki/Ruby\\_on\\_rails](http://pt.wikipedia.org/wiki/Ruby_on_rails)

[10] - <http://en.wikipedia.org/wiki/XML>

[11]-<http://pt.wikipedia.org/wiki/MySQL>

[12] - [http://pt.wikipedia.org/wiki/Oracle\\_%28banco\\_de\\_dados%29](http://pt.wikipedia.org/wiki/Oracle_%28banco_de_dados%29)

[13] - <http://pt.wikipedia.org/wiki/PostgreSQL>

[-http://www.postgresonline.com/journal/archives/51-Cross-Compare-of-SQL-Server,-MySQL,-and-PostgreSQL.html](http://www.postgresonline.com/journal/archives/51-Cross-Compare-of-SQL-Server,-MySQL,-and-PostgreSQL.html)

[14] - <http://pt.wikipedia.org/wiki/HTML>

[15] - <http://pt.wikipedia.org/wiki/JavaScript>

<http://www.wordiq.com/definition/JavaScript>

[16] - [http://pt.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://pt.wikipedia.org/wiki/Cascading_Style_Sheets)

[17]-<http://pt.wikipedia.org/wiki/Scrum>

[18]- <http://sprintometer.com>

[19]- [http://www.sk89q.com/content/2010/04/phpsec\\_cheatsheet.pdf](http://www.sk89q.com/content/2010/04/phpsec_cheatsheet.pdf)